LU TP 19-09

DETECTION OF ARCHAEOLOGICAL SITES FROM AERIAL IMAGERY USING DEEP LEARNING

Author:
Jorge F. Lazo

Department of Astronomy and Theoretical Physics,
Lund University

Master thesis (30 hp) supervised by Mattias Ohlsson

Abstract

In recent years, Deep Learning has proven to be an outstanding tool in the field of computer vision showing promising results in different fields such as the analysis of medical images, obstacle detection for self-driving cars, automatic image caption generation, etc. In the case of Archaeology, the adoption of these methods in the detection of archaeological structures from aerial images has been slower than in other fields. This is an area not widely explored but which seems to have a big potential for the application of Deep Learning methods, given the large amount of airborne data existing.

This work presents the results of an approach using 4 different Convolutional Neural Networks (CNN) models based on different architectures and learning methods. Of the models tested, 3 of them correspond to state of the art pre-trained models for which different techniques of transfer learning were used. The fourth one is a CNN architecture developed specifically for this task. The Deep Convolutional Neural Networks used were trained to carry a binary identification task, in this case, to determine whether an image contains any kind of topographical anomalies corresponding to archaeological structures, or not. The case studies were obtained from the southern Baltic sea region of Sweden and Birka and these correspond to aerial images in the visible light range and infrared. The kind of structures present on the images are burials of different shapes corresponding to the Viking ages.

By using the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) as measurement, the selection of the model best suitable for this task was carried out. Additionally, different augmentation techniques were tried including the generation of images using a Deep Convolutional Generative Adversarial Networks. Finally, an ensemble approach was tested combining the results obtained from the models which showed the best results individually in different types of airborne data. With this approach, a sensitivity of 76% with a specificity of 92% was achieved.

Acronyms

ANN Artificial Neural Network. 2, 12, 18

AUC Area Under the Curve. 22, 24, 34

CNN Convolutional Neural Network. 12, 18, 23, 34, 35, 37–39

DCGAN Deep Generative Adversarial Network. 18, 20, 21, 24, 29, 34, 35

FPR False Positive Rate. 20, 24, 40

GAN Generative Adversarial Network. 18, 35

GIS Geographical Information System. 9

ROC Receiving Operation Characteristic. 20, 24, 26, 34

TPR True Positive Rate. 20, 24, 40

Popular science summary

Several civilizations have flourished and decayed over the short period of time in which humans have inhabited the third planet of a system which belongs to a yellow dwarf star called Sun. This star has been their source of inspiration, hope, questions, and energy and to praise it, as well as just other natural elements, they have built temples.

Many of these constructions have already crumbled, gone forever from the memory of mankind, lost in the haze of ancient times. Some others have withstood the harshness of the weather and other civilizations, remaining in the surface as a remembrance of archaic ages, and as a warning about the fragile nature of human civilization, which sometimes vanishes leaving no trace behind but a pile of rocks. However, there are some others, which have been lost but not forever. Covered by vegetation or new human-made structures they have remained silent, waiting for someone to discover them again and tell the secrets they've kept for their own for so long.

It has been almost 350,000 years of human history, and it is now maybe just the beginning of a new era in which the task of detecting the remains of lost civilizations could be delegated, to some extent, to something else than humans: artificial systems with the ability to analyze a large amount of airborne data, looking for terrain anomalies. So far, the methods used for this purpose rely almost completely on the analysis by a specialist. This is a time consuming task which requires highly specialized knowledge and which may involve interpretation biases.

These artificial systems, which are based on the use of the most astounding programming paradigms: Artificial Neural Networks (ANNs) and Deep Learning (DL), have already proven to be a reliable tool in the field of computer vision showing promising results in different areas such as the analysis of medical images, self-driving cars, automatic image caption generation among many others.

Unlike conventional algorithms, DL and ANN do not have a set of rules predetermined for the system to solve a task since the beginning, instead; given an initial set of data, the system itself finds and recognize the patterns which are useful to make further predictions and analysis. Just as the brain, a single artificial neuron is not capable of all the wonders of which the human mind is capable, but together they can achieve impressive results.

In this project, by using different architectures of a specific kind of an ANN model called Convolutional Neural Networks (CNNs), aerial images in the near infrared and visible light ranges were analyzed to detect graves from the Viking ages. Different methods for improving the performance were tried, including the generation of synthetic data (also using CNNs), and the combination of different architectures with different kinds of data in order to achieve better predictions.

The improvement and research in this methods are of great importance as nowadays there is more data available than the one which is humanly possible to analyze. The detection of archaeological sites is vital, as it allows the retrieval and protection of these structures which can help in the understanding of many aspects of these ancient civilizations, and in general human history, knowledge which otherwise would be lost forever.

Contents

1

Introduction

Light travels through space and reaches our eyes, it enters our optical organ and passes through the cornea, the pupil and then strikes a photoreceptor layer, the retina. The retina is the brain's window to the optical perception of the universe. It is here where the light is phototransducted to neural signals, which will be processed by our cerebral cortex to become what we call visual perception. This amazing process happens all the time, just in front of our eyes, or more accurately right behind them, without us realizing about it, and yet this single process, is something that only nature has achieved to develop to a high level of accuracy, diversity, and robustness to the external conditions. Trying to mimic the human visual system, might be not a recent ambition, but it has been just in the few last years when scientific research in the field of Deep Learning (DL), has come close to making this dream become true.

It turns out an interesting and peculiar case that, the boost in artificial vision came with the adoption of techniques which at its origins had a basis in neuroscience. This is, the implementation of Artificial Neural Networks (ANN') and specifically, in the field of computer vision, the Convolutional Neural Networks (CNNs). The first models and the principles from which this kind of networks are based on, were developed inspired by how the visual information is processed in the brain cortex.

Nowadays, the use of these methods have proven to be an outstanding tool to solve a wide range of problems in computer vision and with practical applications in several fields. Among one of these novel implementations, there is the application in archaeology in different tasks, such as the assistance in the classification of historical pieces encountered during excavations, the retrieval of glass pieces [1] and classification of ceramic vessels [2] or the identification of possible excavations sites by analyzing aerial imagery [3, 4]. About this last task, it is still an area not yet widely explored, and there is still plenty of work to be done as there are many interesting challenges that need to be addressed. In the present work, an approach regarding this application using different deep CNNs models is proposed. A binary classifier, which makes use of aerial images corresponding to two different wavelength ranges to determine if the images contain or not archaeological structures on it, is presented.

1.1 Machine Learning and Image Classification

When it comes to solving analytical and iterative problems, machines and computers have taken over since several decades ago, showing better performances than almost any of the human minds in tasks like doing complex numerical calculations, sorting and finding information in databases or even at playing chess, but still, the most simple and trivial tasks in our everyday life turns into a great challenge for machines. This is in part because the tasks that can be solved by iterative steps can be described in a set of formal algorithms, and the input information for these programs is well defined and understood so it can be processed properly. In most of the activities we carry out in our everyday lives, even if the goals can be well defined, the number of variables and the amount of information we perceive is overwhelmingly high and the attempt to writing an algorithm can turn into a grueling job which may not even lead to the right solutions for all the possible cases.

Machine Learning (ML) arises as a solution to this kind of problems, in which the system itself is capable to determine which is the useful information and which not. Given an initial set of data, the system itself finds and recognize the patterns which are useful to make further predictions and analysis, and all of this is done without specifically programming the system to solve the task. A key element in machine learning is Artificial Neural Network (ANN) which are one of the most astounding programming paradigms and are inspired broadly in how the human brain works. Just as the brain, a single artificial neuron is not capable of all the wonders of which the human mind is capable, but together they can achieve impressive results. ANNs can be coupled together forming layers, and at the same time, these layers can be coupled with other layers, resulting in more interesting models which can achieve striking results. Deep learning can be regarded as a specific case of machine learning [5] where more layers are added to the models, turning them more complex and deeper [6], making an allusion of its own name.

One of the fields which have witnessed tremendous progress just in the last lustrum is the one of image classification, where the main task is to specify a label to an image from a previously defined set of categories. Despite it may sound trivial, the complexity of the problem is stirring because of the big amount of information an image can have. It was not until the last few years when a remarkable improvement in the ways of solving the problem was achieved. In 2012 a Deep Convolutional Neural Network called AlexNet outperformed in the ImageNet Visual Recognition Challenge, achieving an error rate of 16% [7]. Since then, the field has experienced steady progress to the point in which the rate of error of this kind of models, have exceeded human-level performance in single shot classification tasks [8]. It is important to mention that the task of image recognition could vary according to the nature of the problem; broadly speaking some varieties are:

- **Object identification**. One or different classes are identified given an image.
- **Object recognition**. It involves the detection and classification of one or multiple classes along with their position in the image.
- **Image segmentation**. Consists in partitioning the image into multiple segments, or set of pixels, each one belonging to a different category.

1.2 Object Identification and Aerial Imagery

Aerial image interpretation is the process of analyzing aerial imagery with the purpose of identifying objects, and the properties that represent them. The automation and semi-automation of this process has been an active field in research and different techniques have been developed [9, 10, 11, 12, 13, 14]. Some techniques rely on the analysis of low-level and mid-level visual features detection methods, which work with the analysis of e.g. spectral texture and structure of images or the statistical patterns formed by the extracted local visual attributes. In recent years, high-level visual features methods, such as the use of CNNs, have surpassed the performance of the first ones, some comparisons have been carried showing that CNNs have had in general a performance of over 90% of accuracy whereas when using low-level feature analysis, a maximum accuracy of 75% has been reached [11].

Automated object identification from aerial images by itself has been a subject of research since the late 1960s [9], and the techniques and applications of it have evolved together with the development of ML. Between the applications that have been explored, there is the detection of roads and buildings [15, 16, 17], estimation of forest biomass [18], disasters damage estimation [19], detection of poverty and prediction of troublesome areas [20] along with many others. Some characteristics that these different applications share are that these are time-consuming tasks which require to analyze in detail large images; they also require a highly specialized knowledge in order to identify the regions that are really of interest, and there is an interpretation bias when determining if some feature detected in a map is really of interest or not.

Despite the fact that aerial image classification has witnessed significant progress, there are few major issues that hinder the complete development of it; Xia et. al, review these issues in [21], pointing especially in the lack of a comprehensive review of all the methods that exist nowadays and the lack of a proper benchmark datasets to evaluate the performance of this methods.

CNNs permit handling High-Level visual information and achieve respectable performances. It has been reported that by directly using pre-trained models, good results can be achieved in the task of classifying aerial and remote sensing images [13, 12]. In some other cases, pre-trained models have been used as feature extractors for fine tuning [11, 21]. It has also been tried the development of a specific architecture for this purpose from scratch, however, as reported in [22] where the UC-Merced and WHU-RS19 datasets were used to train the model, the accuracy achieved was lower than the ones which make use of the existing architectures and different transfer learning techniques. This can be understood as these datasets comprehend only about few thousands of images and a maximum of 50 categories [10, 23, 24], whereas the big architectures are trained on datasets with few million of images and several thousand of classes. Nevertheless, the digitalization of large datasets in the last years, such as historical maps, the spread in the collection of aerial images from different wavelengths of the electromagnetic spectrum and some other remote sensing techniques which provide of 3D mappings of the terrain, opens the door to the improvement of these models, and makes the task of object identification in aerial imagery a perfect target where DL could be applied.

1.3 Aerial Imagery in Archaeology

In the specific case of archaeology, aerial imagery has been a widely applied tool since the twentieth century. When it was first introduced, it turned out to be a revolutionary tool for the field [25] as it allowed to highlight objects and marks which are barely or not at all visible on the ground level. Nowadays it plays a fundamental role in all stages of archaeological research, not only in the detection but also in monitoring and safeguarding tasks. Regarding the detection of archaeological sites, the current goal, at least for some part of the archaeologist community [1, 26, 27], is the automation or semi-automation of the detection task.

The outbreak of the First World War turned aerial photography into a primary tool in military reconnaissance, and after the war, the techniques developed during those warlike times began to being used for non-military purposes [28], the study of archaeological sites was one of them. The first flight taken for archaeological purposes in Europe was in 1899 in Rome to document the excavations that were in progress in the Roman Forum [29], and since then different technologies and techniques have become part of the repertoire of aerial imaging in archaeology. Nowadays archaeologists do not limit themselves to visible light photographs but also make use of photos taken in different wavelengths, using multispectral scanning sensors and some other tools such the use of radars and LIDAR systems. This last one has lead to exceptional results, being one of the most notable examples the discovery of some ancient Mayan ruins in the Guatemalan jungle [30, 31, 32].

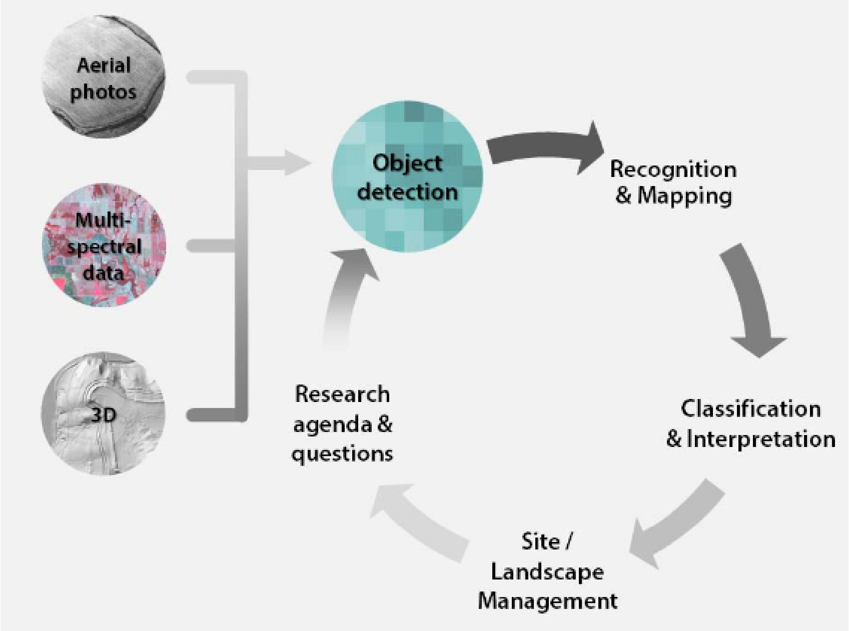

Aerial photos
Multi-spectral data
3D
Object detection
Recognition & Mapping
Classification & Interpretation
Site / Landscape Management
Research agenda & questions

Figure 1.1: The detection and interpretation process. Source: [26]

The detection and interpretation process of aerial imagery in archaeology, consists of several stages as it is shown in Figure 1.1 and even if the detection step is automated or semi-automated, the interpretation and validation of detected anomalies in the terrain from aerial imagery remains a task for archaeologists [26]. The way in which aerial photographs are studied is by analyzing the nature of objects represented in them such as shapes, sizes, shadows, tones, texture, and alike characteristics. The correct analysis of these features can show the residual components of ancient landscapes [29].

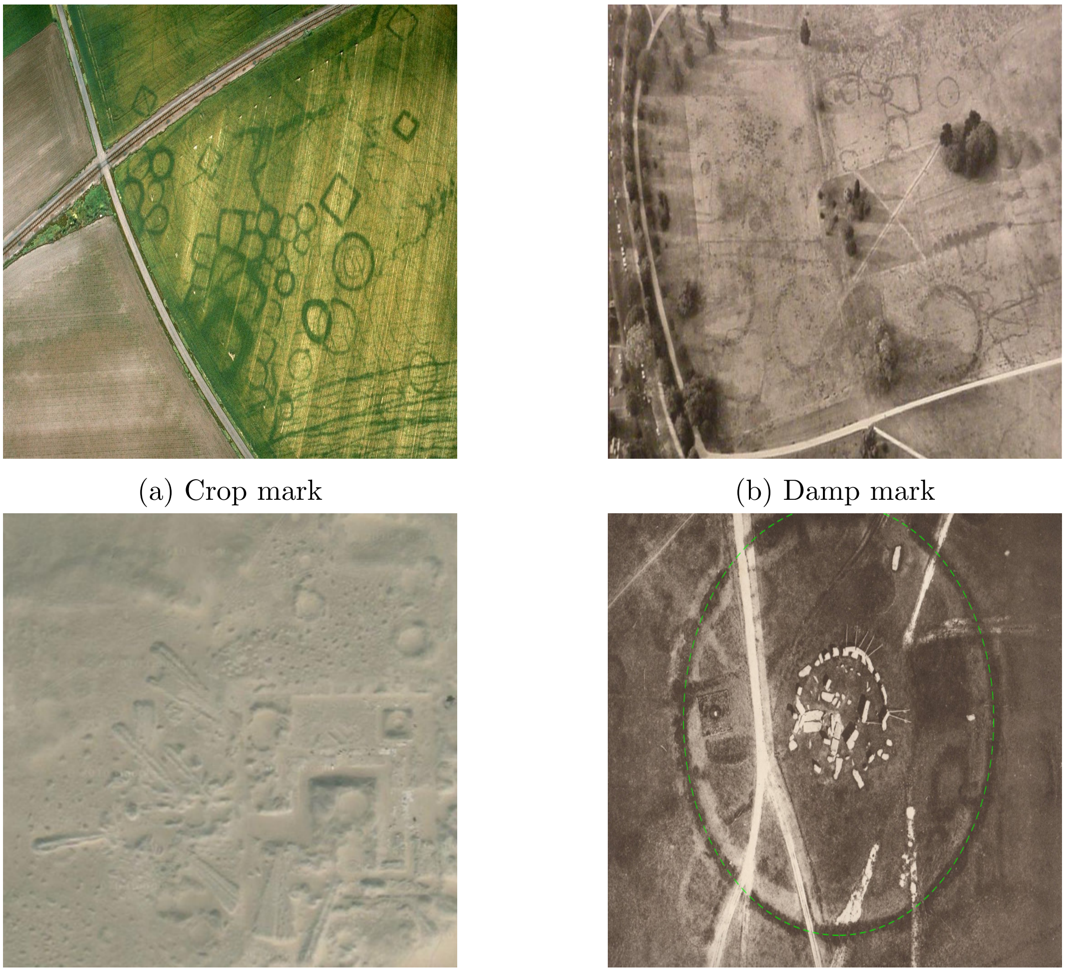
A fundamental part in the detection process, is the correct identification of *archaeological traces*. These are marks which appear in the photographs and are not noticeable by themselves but indirectly when these are compared with the environment that surrounds them. The presence of hidden structures can alter the aspect of the terrain and this kind of traces could be detected thanks to different factors such as humidity, vegetation or the change in the shape and appearance of the ground, some of them are shown in Figure 1.2. As defined in [29, 33], archaeological traces can be of different types, and they are classified as follows:

1. **Damp-Marks**: This kind is seen as a change in the tonality of the terrain; this is because of the fact that soil presents different tonalities depending in the humidity of it.
2. **Crop-Marks**: This kind is similar to the damp-marks, with the difference that the surface has plant coverage.
3. **Soil-Marks**: In this case, the marks have a different color than the surroundings.
4. **Topographical Anomalies**: These type of traces are noticeable in the form of irregularities with the general conditions of the surrounding environment.
5. **Legacy Marks**: These are the kind of marks which have remained on the surface and are easy to verify.

1.4 Aerial Object Identification in Archaeology

The automated and semi-automated identification of sites of archaeological interest seems to be as revolutionary as of how aerial imagery and remote sensing was for archaeology during the last century [25]. A large amount of aerial images and remote sensing data that exists nowadays makes it a perfect target for the application of deep learning techniques, however, it is an area that has not been widely explored yet. Computer vision has already proven to be a trustworthy tool for archaeology in some other areas, such as the use of Structure from Motion (SEM) algorithms to the reconstruction of three-dimensional scenes of excavations zones [38]. The development of better algorithms within the improvement of remote-sensing instruments and techniques has also proven to have a decisive impact in the detection of archaeological sites. Even if in the successful cases, it has not been by using a fully automated processes, it illustrates the potential of applying new methods and techniques in sites detection.

Until now, the number of studies in which machine learning techniques have been applied in for the detection and monitoring of archaeological sites is limited. Menze et al. present an approach making use of the Digital Elevation Module (DEM) data of the Shuttle Radar Topography Mission (SRTM) from the Middle East region using machine learning algorithms. Using a Random Forest classifier operating on an eight-dimensional partial least square (PLS) filter, their method was capable of identifying



(a) Crop mark

(b) Damp mark

(c) Topographical anomaly

(d) Legacy mark

Figure 1.2: Some example images of archaeological traces. Sources:[34, 35, 36, 37] for a, b, c and d respectively

settlement Mounds [3] with a sensitivity of 95.4 percent at a specificity of 98.8%. However, it also presented a high rate of false positives detections, which are mostly due to natural elevations of the terrain that have the same height as the mounds. A subsequent approach by the same author but now additionally using ASTER satellite multispectral images [39] (which indicated the presence of anthrosol), in the area of north Syria, was able to identify two classes: settlements and no settlements from a set of 36 tile images, each one representing an area of 6x6 km [40, 41]. In this approach again a Random Forest classifier was used to perform the feature selection task and resulting in the localization of 14,312 settlements over an area of 22000 km^2 [42]. In a later work the amount of false positives was reduced by removing biased classification maps and estimating the optimal weights for each of the images using a non-negative least square regression strategy [43].

Some other machine learning techniques which have been applied in the investigation of damages in archaeological sites was done by Bowen et al. [44]. Using satellite images from the pyramid fields region of Egypt, a hierarchical categorization and localization (HCAL) algorithm was presented. This method performs, in an unsupervised step, detecting motifs from the set of pictures for training and afterwards categorizing the images according to their features detected using K-means. The members of the similar categories are checked for mismatching and if this is true, the algorithm automatically splits categories again until it has pure categories.

Zingman et al. are one of the few ones who has used pre-trained deep CNNs to detect ruins from aerial images but merely to compare with their handcrafted rectangularity feature detector [45]. In their work they report that their method evaluated in the mountainous regions of the Silvretta Alps presents a better performance than the CNNs architectures tested (AlexNet, Vgg-deep-19, CaffeNet, GoogLeNet and OverFeat) using MATLAB toolbox MatConvNet. They mention that their approach has a notorious lower rate of false positives than the deep learning models. However, no further details are given about the training of the models. Nevertheless, they mention that some of the models tested, using an unbalanced dataset, with few positives and a large number of negatives, may result in a well-performing detector. A similar study presented during the 2017 New Forest Conference, Kramer et. al. [27, 46] suggest that the remarks stated by Zingman may be true. In the presentation of the work it is attested that by using LIDAR data within aerial imagery using the VGG Network pre-trained in Imagenet, the detection system proposed achieves an 85% of validation accuracy.

The most recent studies based on CNNs and transfer learning make use of AlexNet and ResNet18. In the first one, the authors use the network AlexNet as a feature extractor. The images from their dataset are run through the network and then they make use only of the features detected in the last fully connected layer. The features identified, are used as the input for a linear Support Vector Machine (SVM) with which they proceed to the identification of charcoal kilns from the airborne data. Their model was tested in the valley of Lesja in Oppland County, Norway. They compare the true positive rate of detection, false negative and false positive rates with tradittional pattern recognition methods. In the case of the deep learning plus SVM method, the values of the rates reported are 86%, 14% and 37% respectively [47]. In the second study [4], the model ResNet18 was used for transfer learning and implemented as feature extractor; the last layer was substituted with an empty one corresponding to the classes they wanted to detect (Roundhouse, Shieling hut, Small cairn and Background) and then retrained. For their studies they tiled an area of 432 km^2 corresponding to the island of Arran in Scotland in squares of 101x101 pixels. In the case of the ones with archaeological features the structures were centered on the tiles avoiding foreground structures in the learning set. Traditional data augmentation techniques such as rotation, flipping, random translation and random scaling were used. The total amount of images used in the training and validation set was of 9952, and from these 755 correspond to the total amount of classes they were interested in detecting (Roundhouse: 106, Shieling hut: 265, Small cairn: 384). A validation accuracy of 99.46% was reported to be achieved after 4 training epochs. Their method was tested on two areas: Machrie Moor and Glen Shurig, achieving in the first region a 72% of positive identifications for the Roundhouse class and 20% for the small cairn one, however a rate of 87% and 90% of false identifications respectively for each class, was reported. For the second area and the Sheiling hut class, the outcome was a 26% of correct identifications and 189% of false positives.

2

Case Study

The study of graves has been a vital part in archaeology for the understanding of many aspects of different civilization such as their social organization or their conceptions of life and death [48, 49]. Graves and cemeteries from the Viking age provide the most evidence of religious beliefs of pagan Scandinavia, but also about their daily life and society. It is believed that these burials were in some way rituals to conciliate the dead and honor the deceased so it could rest [50]. Boat-shaped burials are believed to have the purpose of supplying the dead with equipment as well as some personal belongings which may be useful for the after-life journey [51].

Two main datasets were provided with 5 case studies corresponding to grave fields in the southern coast of Sweden in the Baltic sea and Birka, on the island of Björkö. The ones corresponding to the Baltic sea area were: Hjortahammar, Hjortsberga and Byrum in Blekinge län and Ängakåsen in Skåne.

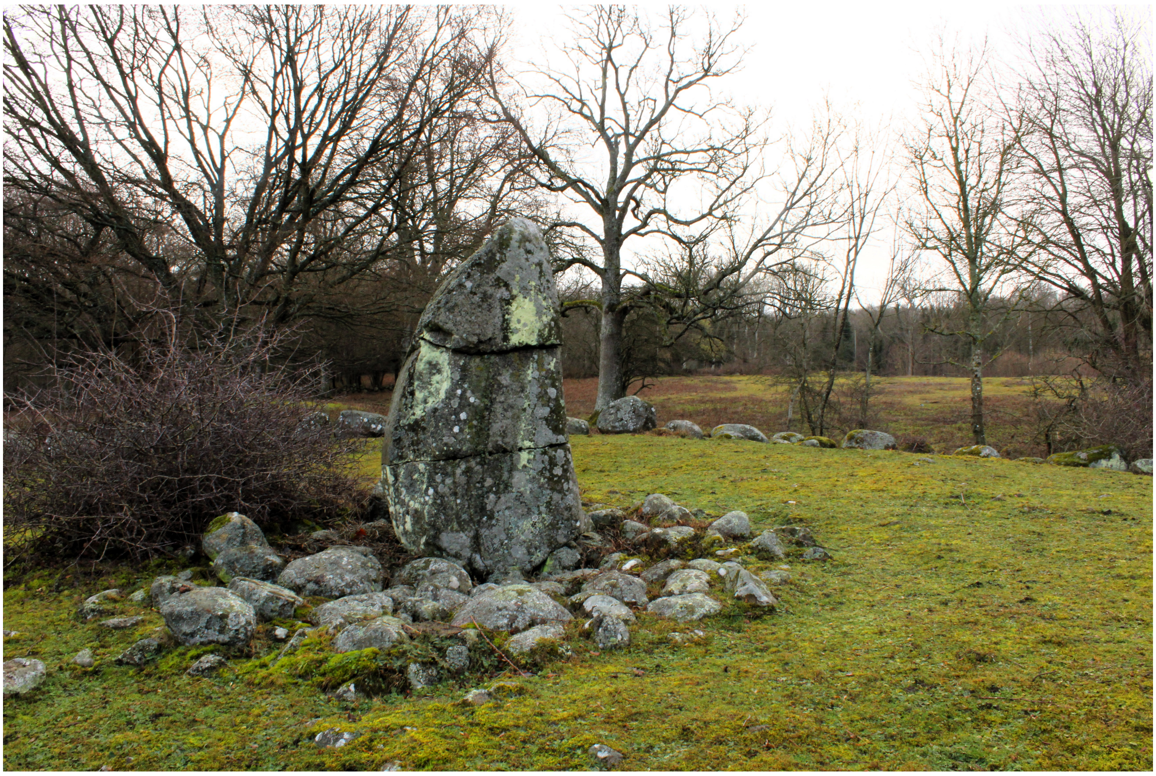


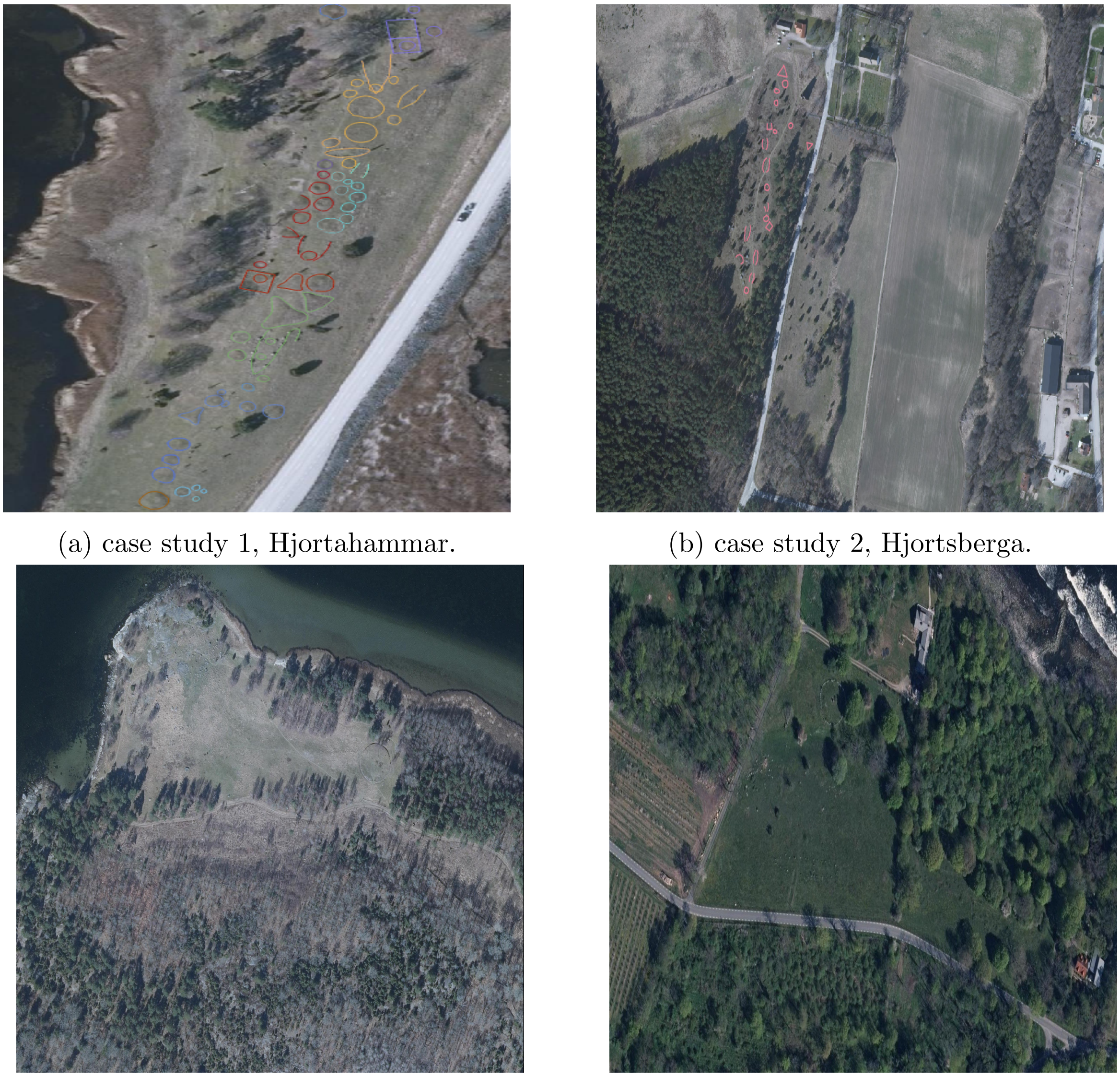
Figure 2.1: Ängakåsen gravfält.

The site of Hjortahammar is situated in the south part of the Johannishusåsen, a gravel ridge surrounded by forested areas up north and the shore of the Baltic sea in the south. The grave field consists of more than 110 iron-age graves dated to the late iron-age. The variety of the graves found in the site are of four main kinds: boat-shaped, triangular, round and four-sided stone settings as well as some erected stones and small mounds. The site has been damaged over the years due to several factors such as visitors and animal grazing, and for this reason it needs to be protected. In order to achieve this, it has been commissioned the creation of an inventory of the graves which includes the mapping of the area using LIDAR techniques and scannings using drones [52]. Ängakåsen is located in Kivik close to Simrishamn and few hundred meters from Kiviksgraven. In this site there is a 60 m long shipping set, two rectangular stone pavements and 130 stone pavements that mark graves, an example of the structures located there is depicted in Figure 2.1. Hjortsberga burial site is located just next to Hjortsberga church in Ronneby municipality, close to the village of Johannishus. The graves in this site correspond to the early Iron Age, i.e. 400 AD. In this zone there are 55 burial mounds, eleven trusses, 13 four-sided stone pavements, five round stone pavements and 19 ship arrangements. The aerial images in the visible light range, corresponding to this datasets from the Baltic sea region, are shown in Figure 2.2.

The site of Birka is located on Björkö island in the lake Mälaren, it is located 30 kilometers west of Stockholm. It is regarded as the oldest town in Sweden and together with the site of Hovgården has been a UNESCO World Heritage Site [53]. The town was founded in the mid 700's and it was an important trading center in Northern Europe during the 9th and 10th centuries. It was the center of the trading network in Scandinavia during the Viking Age. Birka complex is exceptionally well preserved and is one of the most complete and undisturbed Viking settlements [54]. The site has an extension of 226 ha, and the variety of elements in the zone correspond to several monumental mounds, a cemetery, a harbour, a runic stone, among other elements. Of the 3,000 graves existing there, approximately only one third has been studied. The infrared and visible light images of this case study are shown in Figure 2.3.

2.1 The data

The data used in this project was provided by the group of Nicolo Dell'Unto from the Department of Archaeology and Ancient History of Lund University. Each case study was tiled in images of different sizes in Tagged Image File Format (TIFF) with a resolution of 20 cm^2 per pixel. The images provided correspond to the visible light spectrum as well to the near-infrared for some cases. To visualize each of the cases within the tagged shape files of the archaeological traces a Geographical Information System (GIS) software was needed. The software used for this purpose was QGIS [55]. The first 4 case studies correspond to sites in the Baltic sea region whereas the last one to the island of Björkö. The case study 1 is composed of 324 tiles with size of 100x100 pixels, 25 of them with marks of archaeological traces of different kinds and shapes. Aerial photographs in the near infrared, from the same zone were added giving a total of of 648 tiles, with 50 positives in both bands of the electromagnetic spectrum. The case 2 is composed of 285 RGB image tiles corresponding to the visible spectrum, the size of these tiles varied, in some cases to very small sizes (e.g. 12x100 pixels) and without any significant visual information, thus only 269 tiles were chosen to be used, from this batch only 8 tiles have signs of traces.



(a) case study 1, Hjortahammar.

(b) case study 2, Hjortsberga.

(c) case study 3, Byrum.

(d) case study 4, Ängakåsen.

Figure 2.2: Cases study corresponding to the Baltic sea region.
©Lantmäteriet: I2018 / 00119.

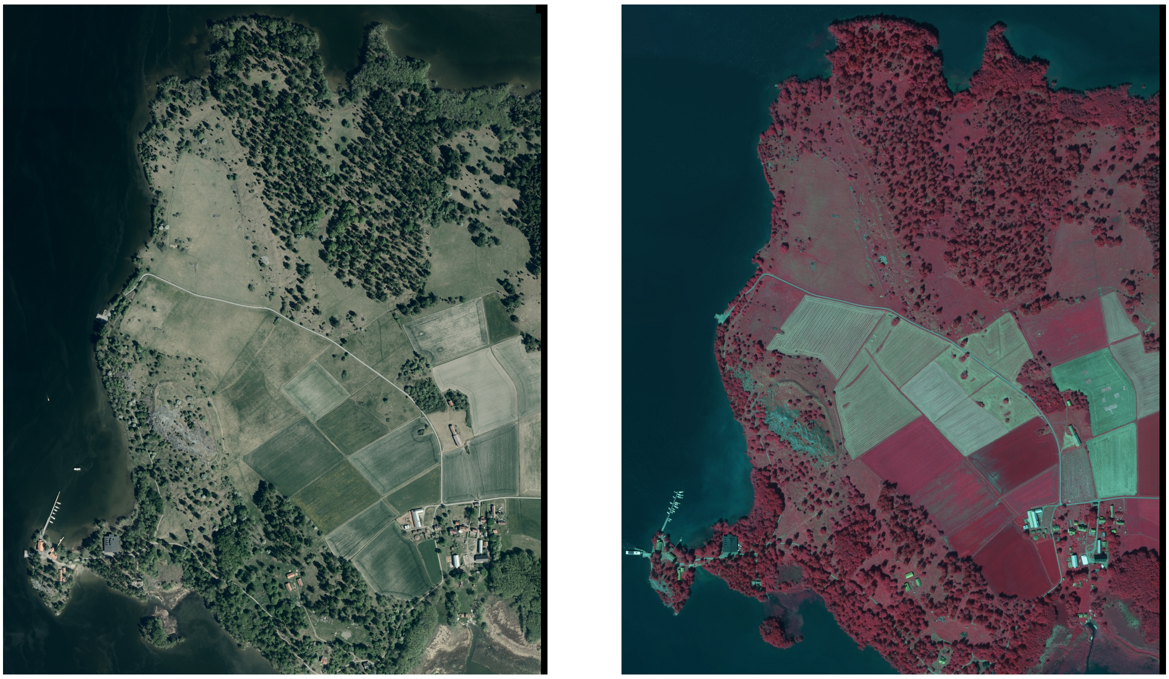
The third case study is composed only by 119 tiles of aerial photographs in the visible spectrum with size 200x200, no tiles with archaeological traces were reported from this batch. Case 4 is formed by a total of 438 tiles in both, visible and near infrared spectrum and 24 tiles reported with archaeological traces. As in case 2 some of the tiles were of very small sizes and without any significant visual information so only 400 images from this dataset were used.

The final case study is from a different region, Birka, and it contains 2726 aerial photographs, with size of 100x00 pixels, in the visible region of the spectrum and the same amount in the infrared. No information about the total number of tiles with positive archaeological traces on it was provided, just the location of them, however a visual inspection showed that this number was of up to 200 for each case. Case 4 was chosen as the test dataset so no further operations were applied to it. The rest of the cases were used for data augmentation, training and validation processes. The composition of the

Case Study	Total number of tiles	Type of data	Tiles with archaeological traces
1	324	Visible + IR	50
2	1055	Visible	32
3	108	Visible	0
4	400	Visible + IR	26
5	5452	Visible + IR	n/a

Table 2.1: Composition of the dataset.

case studies used in this project is summarized in Table 2.1



(a) Picture of Birka in visible light.

(b) Picture of Birka in near infrared.

Figure 2.3: Aerial Images from Birka site.
©Lantmäteriet: I2018 / 00119.

3

Method

3.1 Convolutional Neural Networks

Convolutional Neural Network (CNN)s are a type of feed-forward ANN or Multilayer Perceptrons (MLPs) which are mainly applied in the analysis of images or grid-like structures [5]. The input for this kind of networks could be time series (1D), audio data processed after some operations (like a Fourier transform), images (2D) and volumetric data such as scans or video data (3D). The output, usually, is a list with the score for each class used during the training. This type of deep neural networks have millions of parameters [56, 57] and training these models from scratch requires a large amount of labeled data and also a large amount of computer power. CNNs just as any Deep Forward Network are formed by several hidden layers, but with the advantage that the number of parameters to be self-adjusted is lower, which means a reduction in the amount of training data and computational training time.

CNNs are organized in groups of units called layers, and there are different types of architectures which refers to the overall structure of the network. However it is possible to mention some layers which are present in most of this architectures. The fundamental structure of a CNN consists of three stages; (in the literature sometimes these three steps are counted as a single layer stage, and sometimes each stage is considered separately, in the present work the first option is used) a Convolutional stage, a Detector stage, and a Pooling stage, this is depicted in Figure 3.1.

3.1.1 Convolutional Layers

As its name suggests, CNNs use the convolution operator, which is performed in the first stage of every convolutional layer several times in parallel. When working with images the convolution operation can be defined as:

$$S(i,j) = (I * K)(i,j) = \sum_{m}\sum_{n} I(m,n) * K(i-m, j-n) \tag{3.1.1}$$

Where I is the image as the input and K a two-dimensional kernel. In practice convolutional kernels are small, usually of a size 3x3, 5x5 or 7x7 the biggest. Additionally to the size of the kernel, two other parameters should be defined. The first one, the *stride* with which the filter is slid must be defined, the typical value for this is 1, meaning the

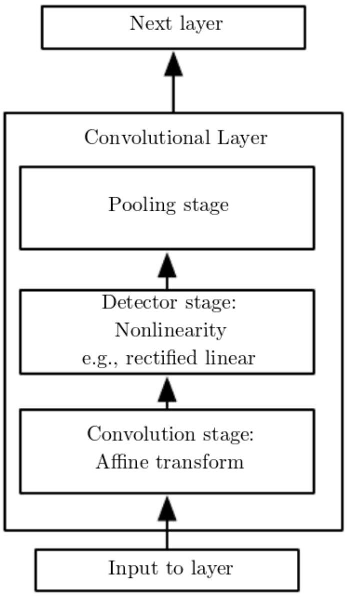


Figure 3.1: Components of a Convolutional Neural Network Layer, Source: [5]

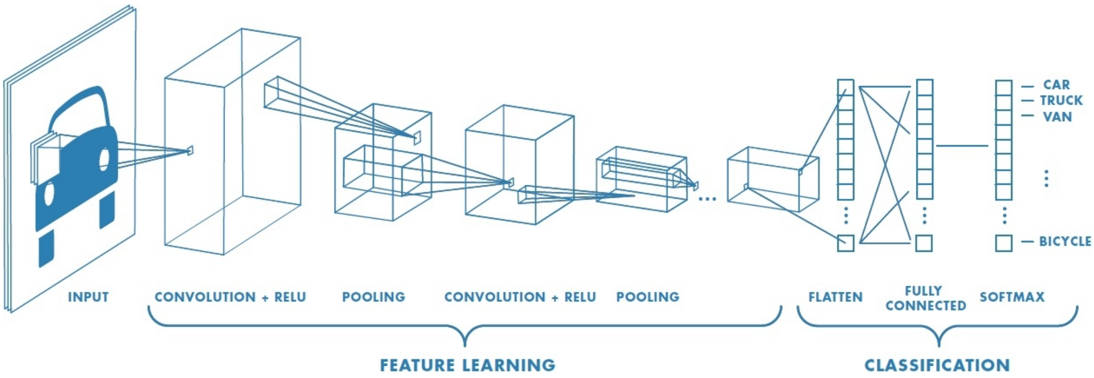


Figure 3.2: Typical architecture of a CNN. Source: [58]

kernel will move 1 pixel at every step, but other values are also possible but rarely used. The second parameter is called *padding* and this refers to padding with zeros the border of the image, but it is not always needed. This parameter is mainly used by architectures which do not make use of the Pooling stage as it allows to control the spatial size of the output.

Unlike standard deep feedforward networks, CNNs weights are shared across an input, i.e. the same kernel weights are used at every position in the input, in this case, the pixels of an image. An easy way to understand it is considering kernels as "*filters*" which are applied to the image, and depending on the values of the kernel it can cause a wide range of effects.

By using different filters, the network will learn which ones are significant to detect relevant visual features such as edges, orientations, colors etc. or more complicated characteristics, in the case of layers in the higher levels. The output from this stage is a 2-dimensional array for each kernel.

The output from the convolution then goes through a set of activation functions in what its called the detection stage which uses a non-linear function, resulting in what is usually known as activation map. The most common operation used here is the *rectified linear function* (ReLU). The operation is defined as:

$$f(x) = max(0, x) \tag{3.1.2}$$

In summary, the convolutional layer has the following hyperparameters;

- The number of kernels n
- The filters size $k_w \times k_h \times d^{(i-1)}$
- The stride s
- The padding p
- The activation function

Where d is the depth or the number of channels, e.g. for RGB images $d = 3$, and the subindexes w and h are the width and the height respectively.

When working with images, the input of a CNN can be considered a volume of size $w \times h \times d$, the same goes for the output of a convolutional layer, where instead of d being the depth, it is now n, the number of kernels.

3.1.2 Pooling Layer

The output from the activation stage goes to a stage called Pooling where a downsampling operation is performed. This operation helps to reduce the spatial size while maintaining the important information, this helps to reduce the number of parameters needed and hence the computation that needs to be performed, however not all the architectures use this stage, some architectures use only convolutional steps, with increase stride to reduce the size of the representation [59]. There are different kinds of pooling which depending on the data and features will perform better than others, still average and max pooling have shown to be the ones which, in general perform better, a graphical representation of both is shown in Figure 3.3. Max pooling is good for tasks in which the separation of features which are very sparse is needed [60].

3.1.3 Fully Connected Layer

This is a layer which can be regarded as a typical multi-layer perceptron and as its names indicates, each neuron in this layer is connected to every neuron in the previous layer. The outputs from the previous convolutional and pooling layers can be seen as the features detected, the fully connected layer makes use of this features to effectuate the task of classification the input to the given classes of the dataset. It is important to mention that the input to the FC layer a flattening step is needed, this turns the pooled feature maps from the previous stage into a long input vector. The *Softmax* function is used as activation function in this layer for classification tasks as it takes a vector of the real-valued scores and give as result a vector of values between zero and one, the sum of all of them should be one.

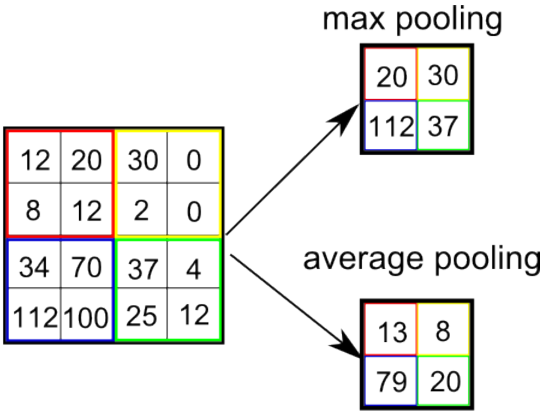


Figure 3.3: Representation of max pooling and average pooling operation in a 4x4 matrix with stride 2.

3.1.4 Hyperparameters

In most of Machine Learning algorithms, there are a set of parameters which can be tuned to control the algorithm's behavior. In the case of CNNs the hyperparameters which can be adjusted to modify the performance of the network when training it are the following ones:

Dropout
It refers to temporarily remove some of the nodes in a network where the removed nodes are chosen randomly, given a probability p to keep the nodes. This technique can be used in any layer except the output layer.

L2 norm regularization

This type of regularization technique consists of adding a constraint c on the node weight vectors, in order to hinder them to grow larger than it. The penalty term is added to the error function which scales with the size of the weights. This induces the network to keep the weights small.

In addition, there are other sets of parameters that are needed to be adjusted before training such as the batch size, the number of training steps and test intervals.

3.2 Transfer Learning

Training a model from scratch requires a large number of computational resources as well as a fair amount of labeled data, it also can take a considerable time (up to 2 or 3 weeks) even using several GPU units [61]. Transfer learning is a method which allows making the process more efficient by using a model which has already been trained, and depending on the task there could be different scenarios for transfer learning:

- **Feature extractor**. In this case, only the last layer is retrained to adjust to the new dataset, and it makes use of the features that have already been detected in the original data set.

- **Fine tuning**. In this case, not only the last layer of the CNN is retrained, but also the weights of the network (not necessarily all, in most of the cases only the high-level ones in the last layers of the network) [62].

3.3 Data Augmentation

In many cases the data set available may not generalize to all the conditions, making the system to overfit on the training data [57]. In the case of images, the conditions of a certain class depicted in the picture can change in many ways such as translations, size, illumination etc. thus, it would be desirable to have a model which is invariant, i.e. a model which can make classifications regardless of these changes. The way to solve this kind of difficulties is by creating synthetic data, which covers a broader set of conditions but preserves the fundamental labeled information, this is called data augmentation. By doing this, it is expected that the network can generalize during the training the truly significant features and perform correct classifications in a different dataset from which it was trained.

Data augmentation can be done in two ways, offline or on the fly. The first one refers to enlarge the data set before performing the training of the network, and this option is usually taken when the original data sets are small. The resulting data set will have a size proportional to the number of operations performed in the original one. The second way of performing data augmentation is on the fly; in this case the operations are carried out as the data, rearranged in mini-batches, is feed to the model for training. This kind of data augmentation is preferably used for large data sets.

When it comes to data augmentation in images, there a several techniques that can be used, some of them are shown in table 3.1:

Type of Augmentation	Description	Augmentation Factor
Flip	Flipping the images horizontally and vertically	2
Rotation	When the orientation of the class is not vital, the image can be rotated at any angle, however this may lead to a change in the size of the image after rotation, so a re-scaling operation may be applied afterwards.	3 $(\delta = 90 \in [0, 360]$)
Brightness	It can be defined as the relative amount of energy output from a source light to another reference point.	~ 20 $(\delta = 0.05 \in [0, 1]$)

Type of Augmentation	Description	Augmentation Factor
Contrast	It is defined as the difference of intensity between the maximum and minimum value of the pixels in an image. This difference can be increased or reduced randomly.	~ 20 $(\delta = 0.05 \in [0, 1]\ )$
Saturation	It refers to the amount of a "pure" color in an image, it can be represented as the amount of gray in proportion to the hue.	~ 20 $(\delta = 0.05 \in [0, 1]\ )$
Hue	Hue can be seen as the "shade" of the colors in an image. The strength of the effect can be adjusted randomly in the whole picture.	~ 50 $(\delta = 0.02 \in [0, 1])$
Relative luminance	It is described as the amount of light transmitted or reflected in a particular area, i.e. the radiant flux density weighted by the luminosity function (the average sensitivity of human visual perception of brightness).	~ 18 $(\delta = 0.05, \in [0.6, 1.5]\ )$
Noise	A neural network could go trough overfitting when it tries to learn features that appear with high frequency but which are not useful. By adding the right amount of noise, e.g. gaussian noise, the high frequency features are distorted, and then the performance during the training can improve. This is also helps to add noise robustness to the system.	~ 50 $(\delta = 0.02 \in [0, 1]\ )$

Table 3.1: Overview of augmentation techniques. The augmentation factors were calculated for typical situations. δ is the size of the difference between different values in the ranges stated.

In some cases, the techniques previously stated cannot be generalized to the variety of conditions in which natural data could exist, as it is not portrayed in the data available for training. Nonetheless, this is not an obstacle as nowadays there are some techniques which can solve this issue, such as the use of Generative Adversarial Networks[63], or the use of Autoencoders [64].

3.3.1 Generative Adversarial Networks

Generative Adversarial Network (GAN) are a type of generative models which learn to capture the statistical distribution of some input data, allowing to synthesise samples from the learned distribution. This is achieved during a competition process inspired in a game-like scenario between two networks, the *generator* and the *discriminator*. The first one tries to create realistic samples while the discriminator tries to discern if the generated data is real or not. Since its first appearance [65] many other models based on its working principle have been proposed and these have been applied in different tasks such as the generation of images, image editing, style transfer and even for fast-simulation of particle detectors and modeling of particle cascades [66].

One of the main issues with conventional data augmentation techniques is that in some cases the data generated generalizes poorly. Some recent work has shown that GANs improve the performance of image classifiers systems and it turns particular useful when the amount of data of the class which need to be classified is limited [63]. Research in this area has also shown that GANs are suitable for representation learning [67]. In the specific case of aerial imagery GANs are an active area of research, some of the applications which have been published with this type of networks are the prediction of ground level layout from aerial images [68], the image to image translation from maps to aerial images [69, 70], or the way around from aerial images to maps [71].

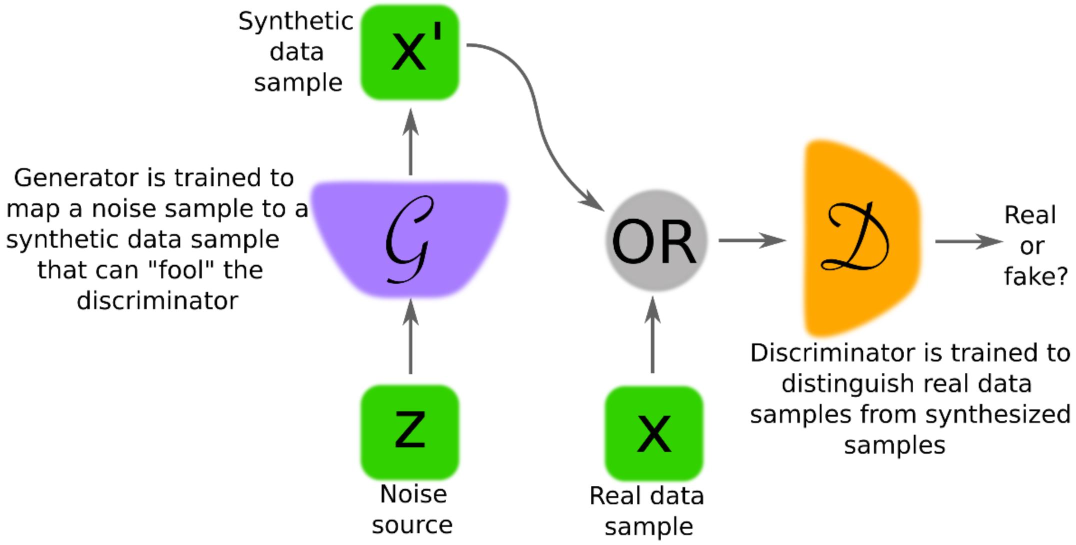

Synthetic data sample
X'
Generator is trained to map a noise sample to a synthetic data sample that can "fool" the discriminator
G
Z
Noise source
OR
X
Real data sample
D
Real or fake?
Discriminator is trained to distinguish real data samples from synthesized samples

Figure 3.4: Graphical representation of a GAN network, the discriminator D and the generator G are usually implemented using ANNs. Image source: [67]

The GAN model used in this project to generate images was based in the Deep Generative Adversarial Network (DCGAN) architecture proposed in [72]. This model does not makes use of a standard CNN architecture, as it does not make use of pooling stages, instead it applies transposed convolutions which upsample (generator) or downsample (discriminator) the previous layer as if a pooling operation would be applied. The *ReLU* activation function is used in the generator after every convolutional layer and in the output layer the *tanh* function is used. In the case of the discriminator the leaky rectified activation is used in every layer.

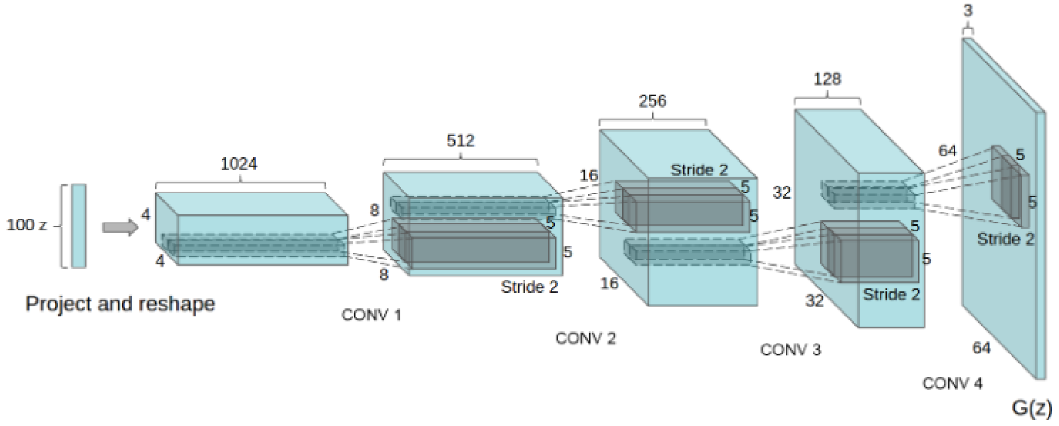


Figure 3.5: Architecture of the Generator used to produce aerial images with archaeological traces on them. Source: [72]

3.4 K-fold cross validation

Cross validation is a sampling method which is useful to evaluate models when the amount of data is limited. In this procedure the parameter k indicates the number of groups in which the total data used will be divided, the use of it in general allows to obtain results less biased, as each model is tested in different sets of data. When the dataset is divided is important to take into account that the data is evenly distributed in each of the k datasets.

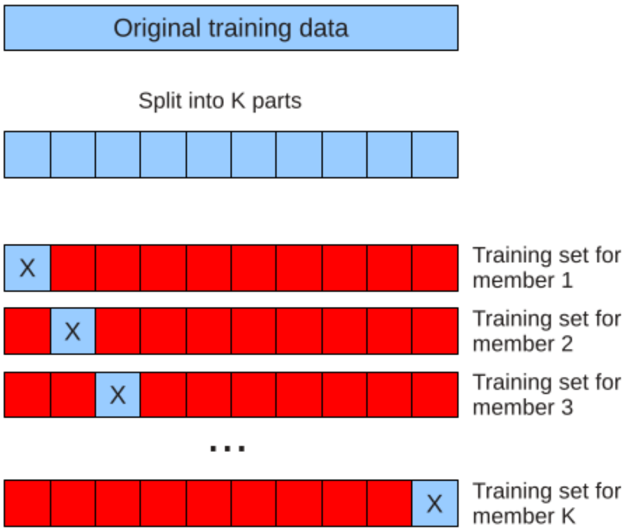

Original training data
Split into K parts
X
Training set for member 1
X
Training set for member 2
X
Training set for member 3
...
X
Training set for member K

Figure 3.6: Graphical representation of K-fold cross validation.

4

Implementation and Results

Given the nature of the dataset obtained for this task (see table 2.1), in which there are only a total of 108 sample images with archaeological traces, the classification task was chosen to be carried out in a binary way despite the fact that the images with archaeological structures belong to different kinds of traces and different shapes. It is also important to mention that because of the way in which the tiling of the aerial images was done (dividing the whole aerial images uniformly in squares of constant size despite their content), in some of the cases the archaeological structures appear divided in different tiles, and in some other cases there is more than one single structure in the tiles.

The classification models were developed to determine whether an image contain any kind of archaeological structure, of the ones existing in the training dataset, or not.

4.1 Performance Measurements

When working with classification tasks involving neural networks, the most commonly used performance measurement is the accuracy and loss, these ones are usually displayed against another hyper-parameter such as the epoch, however, depending on many facts such as the specific task, the intrinsic characteristic of the dataset etc. in some cases these may not provide a proper evaluation of how the system is performing.

For the case of the DCGAN, used to generate images for the training dataset, the accuracy was the performance measurement used for the training and validation process. For the specific classification task of this project, in which the class that we are more interested in detecting represents barely the 5.7 % of the total data (see section 2.1). A system which would classify all the image tiles in the set as negatives, could reach an accuracy of over 94% meaning that this performance measurement is inefficient. An alternative to it is the use of ROC curve.

The Receiving Operation Characteristic (ROC) curve is a graph that shows the ability of a classifier to perform a prediction as a threshold parameter is varied. The curve is the result of plotting the True Positive Rate (TPR), also know as sensitivity, against the False Positive Rate (FPR), defined as

$$TPR = \frac{TP}{TP+FN} \tag{4.1.1}$$

$$FPR = \frac{FP}{FP+TN} \tag{4.1.2}$$

where TP are the true positives, TN the true negatives, FP, the false positives and FN the false negatives. The *specificity* is related to the FPR as $1 - FPR$. ROC curves, are particularly useful for selecting classifiers based on their performance [73].

The Area Under the Curve (AUC) is simply the area under the ROC curve and it is a measure of the discriminability of a pair of classes.

4.2 Data augmentation results

Data augmentation was performed in two ways: using typical augmentation techniques showed in Table 3.1 and by generating images with archaeological structures using a DC-GAN based method (see section 3.3.1). Each of them was tested separately. The case study 4 was chosen as test dataset so no further operations were carried out in the image tiles from it. This left the training and validation dataset with only 82 images containing archaeological structures, to deal with this, some images with archaeological structures from case study 5 were added to this dataset. The composition of the dataset used for data augmentation was of 207 images containing archaeological structures and 1405 without.

An advantage of the aerial imagery type of data is that the features which we were interested in detecting do not depend on the orientation, giving a wide margin to apply image transformation methods. As the dataset is not very big, it was decided to apply offline-data augmentation (see section 3.3).

The way in which the first data augmentation dataset was obtained was by first applying the rotations, and then flipping them vertically, this gave a total of 8 different images. The final step was to choose randomly two of these 8 images and again randomly choosing two of the other 6 operations to be applied on the two chosen ones. The dataset was augmented by a factor of 10, having a final composition of 2070 images with positive traces and 14050 without. This was the dataset with which all the subsequent tests and experiments were performed.

In the case of the method based on the use of a DCGAN model, it was trained for 250 epochs, taking as the real data samples the 82 images reported to have archaeological structures. The system was trained for 2500 epochs obtaining a final accuracy of 0.9997 and generating a total of 52 synthetic images which were added to the positive class. Some of the images generated are shown in the appendix D. The attempt to select a model using this dataset did not succeeded so it was used only during the test stage once the final selected models were already tuned.

4.3 Model selection

In this project a 4-fold cross validation was used in the augmented dataset based on classical techniques. The split of the data was done after data augmentation with each different subset containing around 510 images of the class with archaeological structures.

The split during the k-fold process was carried out separately on each of the two classes by randomly choosing one of the images and then randomly moving it to one of the new 4 datasets. Once this procedure was done one of the k groups was chosen to be the validation set whereas the remaining 3 were chosen as training set. In the final step an evaluation on the test dataset, which did not belong to the k-fold dataset, was performed.

Four different models were implemented and optimized to detect the desired class. Three of the models were based in already existing architectures and one was developed from scratch. The pretrained models used were Inception V3 [74], VGG16 [75] and ResNet50 [76], all of this models have been trained on the dataset *ImageNet* which is an image database with more than 100,000 classes and with an average of 1000 images for each class [77]. It is important to notice that each of them have different architectures and in the case of Inception V3 and ResNet50 they make use of different convolutional modules than the ones presented in section 3.1. A further explanation of these modules and a brief description of the architectures can be found in the appendix A.

For each pretrained model fine tuning and feature extraction techniques were tried. The first one was done by changing the last fully connected layer for one with the total number of classes which were intended to be classified, in this case 2, and then fine-tune different number of layers in the network. The layers chosen to be fine-tuned were from the very last one up to the last 3 layers of each network. In the case of the feature extraction method, an additional convolutional block was added and trained, this was composed by a Global Average Pooling Layer, a Fully connected layer with *ReLU* activation function and dimensional output size x, which was determined using cross validation, and finally a fully connected layer with *softmax* activation function and size equal to the number of the classes desired to detect. All of the different training methods were run using Adam optimization algorithm with a learning rate of 0.001.

Taking as performance measurement in the validation dataset the Area Under the Curve (AUC), it was determined which option from the transfer learning methods, fine-tuning or feature extraction, was the best for all of the 3 different pretrained models tested. Fine-tuning the last layer showed the best results in the validation stage, however further analysis in the test dataset resulted in poor performances so the option was discarded. The feature extractor option which did not show impressive results in the validation dataset, but still fairly good ones was further explored for each of the architectures. In this case, an addition of a Dropout module just before the Global Average Pooling Layer and L2 norm regularization in the fully connected layer was tried. The values for these last two options were randomly chosen and again using k-fold validation the one with the best results was chosen for the proceeding tests. Only in the case of ResNet50 adding dropout showed to have better results than by not doing it, the value selected for it was 0.5. The size of the output for the fully connected layer was 1024, 2048 and 256 for Inception V3, VGG16 and ResNet50 retrained models, respectively and the L2 norm regularization value for which the models had a better performance were: 0.001 for VGG16 and ResNet and 0.008 for Inception.

An alternative method, following the approach presented in [78] where a CNN model is proposed to classify ceramic shards was carried out. Even though it has been shown that the performance of deep CNNs is correlated to the number of layers in the network, creating deep networks is not as simple as adding layers [79] and currently there is not a formal theory which can tell how to choose the optimal number of parameters, such as number of layers, size and number of kernels per layer, learning rate, etc. In the case of the CNN model developed it was initially based on [9], and then additional convolutional blocks were added; as proposed in [75] it was chosen to have a fixed size of the kernels in every Convolutional and Max Pooling layers. The final number of convolutional blocks (see figure 3.1) was determined using k-fold validation, each of the different k-models were trained in a CNN with different number of convolutional blocks. For every block the ReLU activation function was used, except in the last one were the *softmax* function was implemented. The size of the kernels for all the blocks was set to 3 followed by a MaxPooling operator with stride 2 and pool size of two. The final composition of the proposed CNN is shown in Table 4.1.

Table 4.1: Architecture proposed.

Layer	Details	Parameters	Output size
Input	Image	-	100x100x3
Conv2D	512 kernels size = 3x3	14336	98x98x512
Activation	*ReLU*		
MaxPooling2D	pool size = 2x2, stride=2	0	
Conv2D	256 kernels size = 3x3	1179904	47x47x256
Activation	*ReLU*		
MaxPooling2D	pool size = 2x2, stride=2	0	
Conv2D	128 kernels size = 3x3	29504	21x21x128
Activation	*ReLU*		
MaxPooling2D	pool size = 2x2	0	
Conv2D	64 kernels size = 3x3	73792	8x8x64
Activation	*ReLU*		
MaxPooling2D	pool size = 2x2	0	
Conv2D	256 kernels size = 3x3	295168	2x2x256
Activation	*ReLU*		
Dropout	value = 0.5		
MaxPooling2D	pool size = 2x2	0	
Flatten	-		
Dense	256	256	65792
Activation	*ReLU*		
Dense	2	2	514
Activation	*softmax*		

Output

4.4 Results in the test dataset

Once the optimal values were found for each of the models, they were tested in the case study 4 by making predictions using the trained networks. The predictions were carried for each of the two types of aerial images available (infrared and visible light) separately. This final training and testing procedure was done with 3 different datasets, the original dataset, the dataset using classical data augmentation and the dataset which contained the images generated using the DCGAN network. For all of them, the ROC curve was plotted and the AUC calculated.

As it is possible to see from the ROC curves in Figure 4.1, two of the models (VGG16 and Inception V3) present promising results, in both cases a TPR of 0.8 is obtained before reaching a FPR of 0.2 regardless the dataset in which they were trained. In the case of ResNet50 the graph suggest that the network has gone in over-fitting as the model obtain scores of AUC over 0.91 in the validation dataset, however its performance in the test dataset goes down to values of 0.605 and 0.864 in the best cases. Further investigations in the methods of tuning the parameters of this network are needed. For the model proposed a similar behavior in its performance is observed, again the scores in the validation dataset showed to be a reliable model, having an AUC value above 0.9. By looking at the graphics and the predictions obtained with this network, depicted in the images in the appendix C, it seems that the network has learned to detect very specific features of the training sets which are not generalizable. This results in bad predictions, like cropland and bushes, and it gives a hint of which were the features learned from the training dataset, such as round shapes and contrasts between the foreground and background objects. This may suggest that the network is not deep enough, as it can detect akin characteristics but cannot discriminate the specific ones which correspond to real archaeological structures from similar objects.

Even if the ROC curves show the point at which certain TPR value is reached against the value of certain FPR, they do not tell us which is the threshold value for which this specific rates were reached. In the majority of binary classification systems usually the threshold to determine whether an image belong to the desired class or not is set to 0.5. However, for the case of the architectures tested separately, this “threshold value” may not be the best choice. An investigation on the variation of this number was done. Additionally, given that the location of each of the different archaeological structures was provided for the case study 4, the reckoning of the number of detections for each of the different archaeological shapes existing in this case study (ship, circular and square) was done. By doing this it was also intended to determine which ones were easier and harder to detect. It is also important to notice that as mentioned before in the section 2, because of the way in which the tiling was done, some of the structures were split in different images and in some other cases more that one single structure appears in one image. The results shown in table 4.2 mean the number of tiles in which a structure or a part of that kind of structure was detected, not the number of structures detected, actually the total number of structures existing in case study 4 is six.

The threshold values chosen were $c \geq 0.5$, just as a typical classifier, $c \geq 0.3$ and $c \geq 0.1$. The results for the case in which data augmentation was used are shown in table 4.2. A graphical representation of them is depicted in the Appendix C.

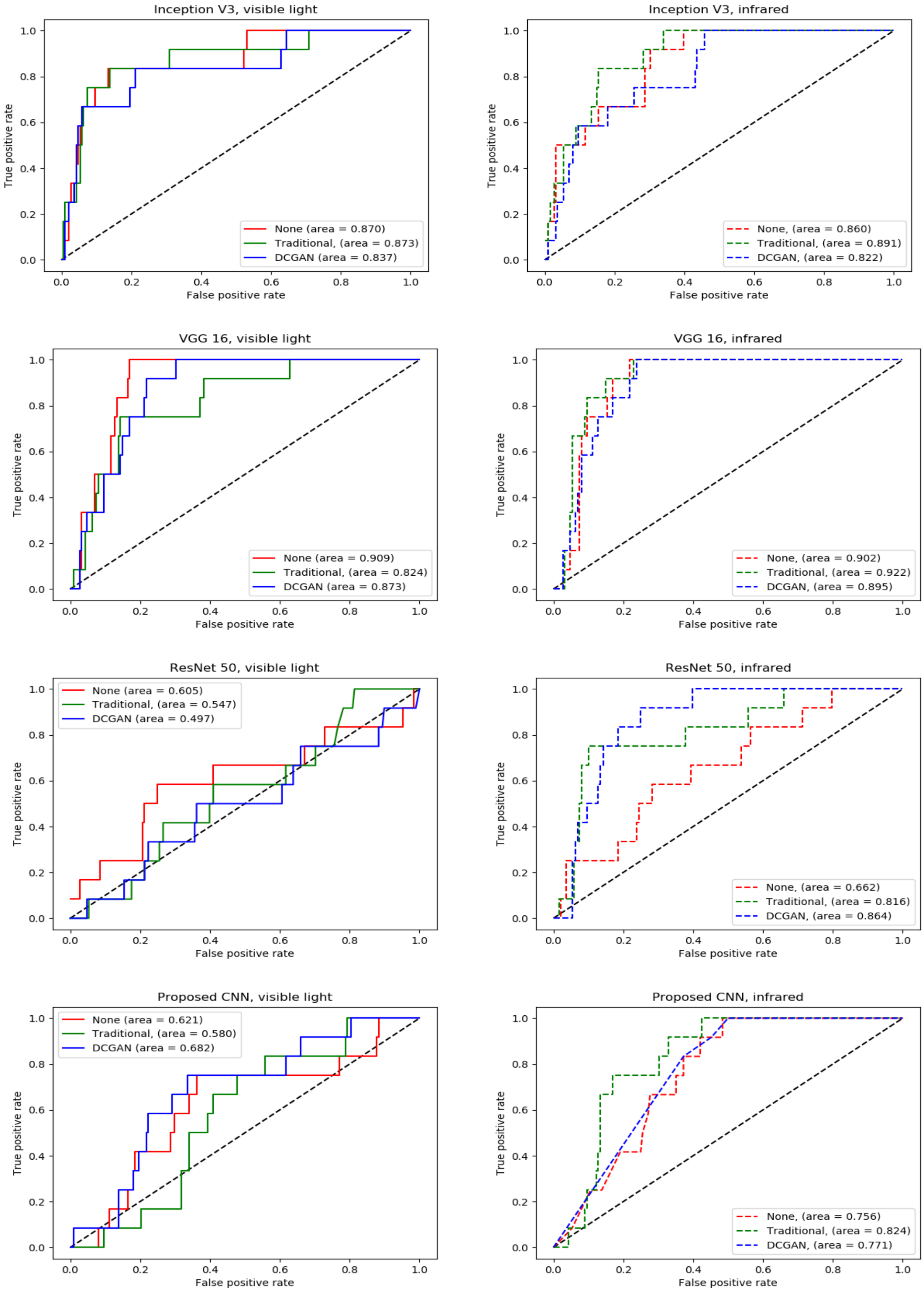


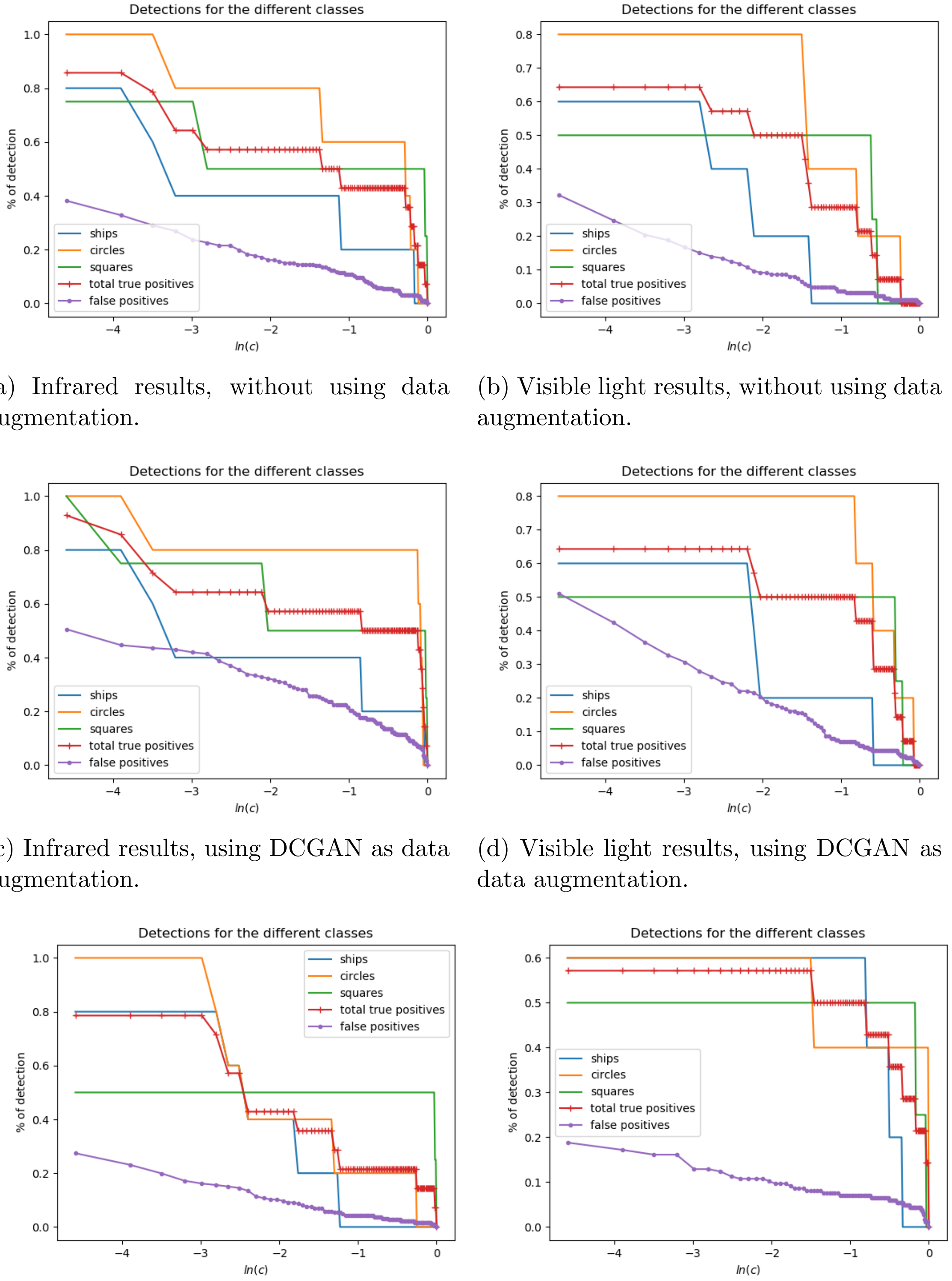
Figure 4.1: ROC curves obtained for the test dataset using for the training-validation process different datasets (no augmentation, DCGAN method and typical data augmentation) in different models. Left column: data from visible light images. Right column: near infrared images. The models used, from top column to bottom column: Inception V3, VGG16, ResNet50 and the proposed CNN.

	Type of Structure	detected, $c > 0.5$		detected, $c > 0.3$		detected, $c > 0.1$	
		Infrared	Visible	Infrared	Visible	Infrared	Visible
Inception V3	Ship	0	2	0	3	2	3
	Circular	1	2	1	2	2	2
	Square	2	2	2	2	2	2
	Total true positives	3	6	3	7	6	8
	False positives	6	12	9	14	21	20
VGG 16	Ship	4	0	4	0	4	1
	Circular	4	1	5	1	5	2
	Square	4	0	4	2	4	2
	Total true positives	12	1	13	3	13	5
	False positives	40	6	57	11	100	25
ResNet 50	Ship	0	0	0	0	4	4
	Circular	0	1	1	1	5	5
	Square	0	0	0	1	4	4
	Total true positives	0	1	1	2	13	13
	False positives	0	6	9	9	187	187
Proposed CNN	Ship	4	0	4	0	5	0
	Circular	5	0	5	0	5	0
	Square	4	0	4	0	4	1
	Total true positives	13	0	13	0	13	1
	False positives	168	18	168	25	170	38

Table 4.2: Tiles detected using classical data augmentation for different threshold values. The number of detections for each shape correspond to the tiles detected in which a structure or part of a structure of this type is contained, not the number of structures detected.

As it can be seen in Table 4.2, the rate of false positives for VGG16 and Inception V3 is low up to a threshold value of 0.3. This suggest that even if the models might not be that efficient at detecting archaeological traces, they are good at determining the aerial landscapes which definitely do not correspond to archaeological structures. This shows that in 90% of the tiles which correspond to the negative class, the probability assigned by the classifier of belonging to the positive class is below to 0.1, or what is the same, they are classified with a probability above 0.9 of belonging to the negative class.

A more detailed exploration was carried out to see how the percentages of true detections and false positives change for $c < 0.1$. In this case the value c was varied from 0.001 to 1 in steps of 0.001 and at each step counting the number of detections for each type of structure. The results obtained using InceptionV3 and VGG16 are shown in Figure 4.2 and Figure 4.3 respectively. The results from these figures provide valuable information to understand how the use of different data augmentation techniques during the training process can affect the ability of a system to make predictions, something that seems hard to distinguish from the ROC curves. Furthermore we can compare how these two models behave to it.



(a) Infrared results, without using data augmentation.

(b) Visible light results, without using data augmentation.

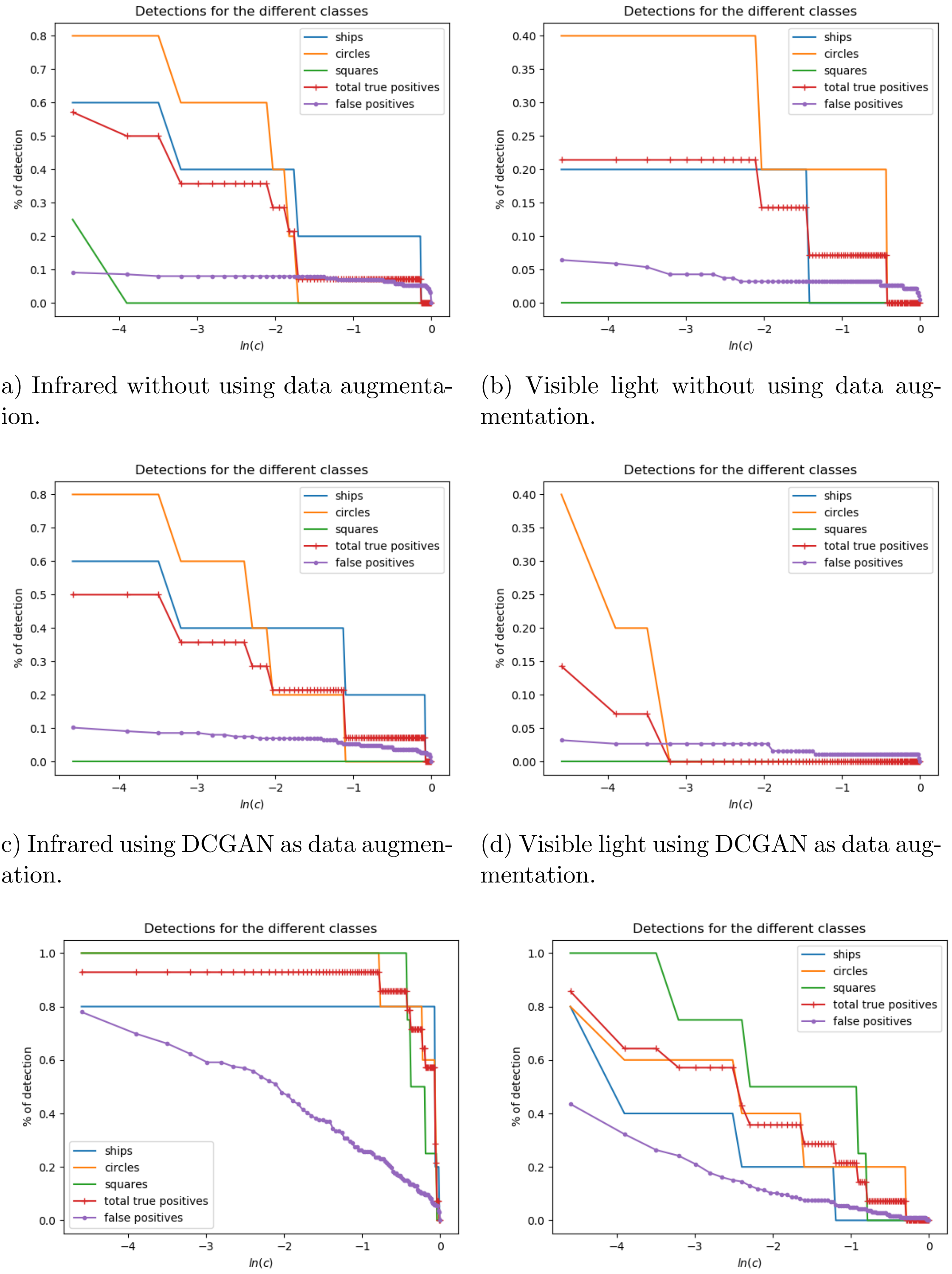
(c) Infrared results, using DCGAN as data augmentation.

(d) Visible light results, using DCGAN as data augmentation.

(e) Infrared results, using classical data augmentation.

(f) Visible light results, using classical data augmentation.

Figure 4.2: Percentage of detections by changing the threshold parameter c. The model was trained using transfer learning with Inception V3.



(a) Infrared without using data augmentation.

(b) Visible light without using data augmentation.

(c) Infrared using DCGAN as data augmentation.

(d) Visible light using DCGAN as data augmentation.

(e) Infrared using traditional data augmentation.

(f) Visible light using traditional data augmentation.

Figure 4.3: Percentage of detections by changing the threshold parameter c. The model was trained using transfer learning with VGG 16.

The first characteristic to notice is that when no data augmentation is implemented, in general, Inception V3 has a better sensitivity than VGG16, but the rate of false detections is also higher, and as the threshold parameter is lower, the number of false negatives increases faster. On the other hand with VGG16 the percentage of false positives is steady, keeping a value below 0.1% even when c is in the order of e^{-5}, however the sensitivity is also poor. This suggests that without data augmentation VGG16 is really not learning but classifying almost all (if not all) the tiles as negatives, which still would mean a high value of accuracy whereas Inception V3 is learning insufficiently. When the synthetic images, generated with the use of the DCGAN, are used as data augmentation method the results does not seem to change so much for neither of the models, just a slight improvement for Inception in the detection of the circular structures, which may be related on the appearance of the images generated (see Appendix D).

The best improvement is observed when data augmentation, using typical techniques, is used to train the models. The performance of VGG16 for both kind of images rises significantly, but in the case of the infrared images, the classification improves remarkably, reaching a sensitivity of almost 100%, however the rate of false positives increases as well to a value above 30%. In the case of Inception V3, the use of data augmentation seems to lead to better results in visible light images.

In the case of the detection of different structures, it was not found any shape which was easier to detect, but as it is possible to see from figures C.2, C.3, C.4 and C.5, the structures which are even noticeable at ground level, were easily detected by the different networks. Furthermore, many of the false detections obtained were of images which could be called "*controversial*", as these contained rocks, or set of rocks which resemble this kind of archaeological structures. Some examples of real detections, false positives and false negatives are shown in Figure 4.4.

By looking at the numbers in table 4.2 and the figures 4.2 and 4.3 it is possible to reaffirm that the model which shows a better performance in the prediction of images with archaeological structures in the infrared is VGG16 while in the visible light range, the one which showed the best performance is Inception V3. In all of the cases, the use of typical methods of data augmentation improved the performance of detection in the system, however not in a consistent way for both models, i.e. the improvement is not the same in both wavelengths range. A final test was carried out, in this the results obtained for each wavelength were combined according to the set from which the system was trained. By doing this the detection rate was enhanced and thus it was only considered the classification output with a threshold value $c \geq 0.5$, the results of each combination for VGG16 and Inception V3 trained on the augmented dataset are shown in table 4.3 and 3 of the images obtained are shown in figures 4.6, 4.7 and 4.8.

Model (data)	% true positives	% false positives
Inception V3 (Infrared+Visible light)	0.38	0.01
VGG16 (Infrared+Visible light)	0.69	0.08
VGG16 (Infrared)+Inception (Visible light)	0.76	0.04
Inception V3 (Infrared)+VGG16 (Visible light)	0.23	0.01

Table 4.3: Percentage of true positives and false positives detection for the different combinations of VGG and Inception V3 trained on the data augmented dataset and taking as threshold value $c = 0.5$.

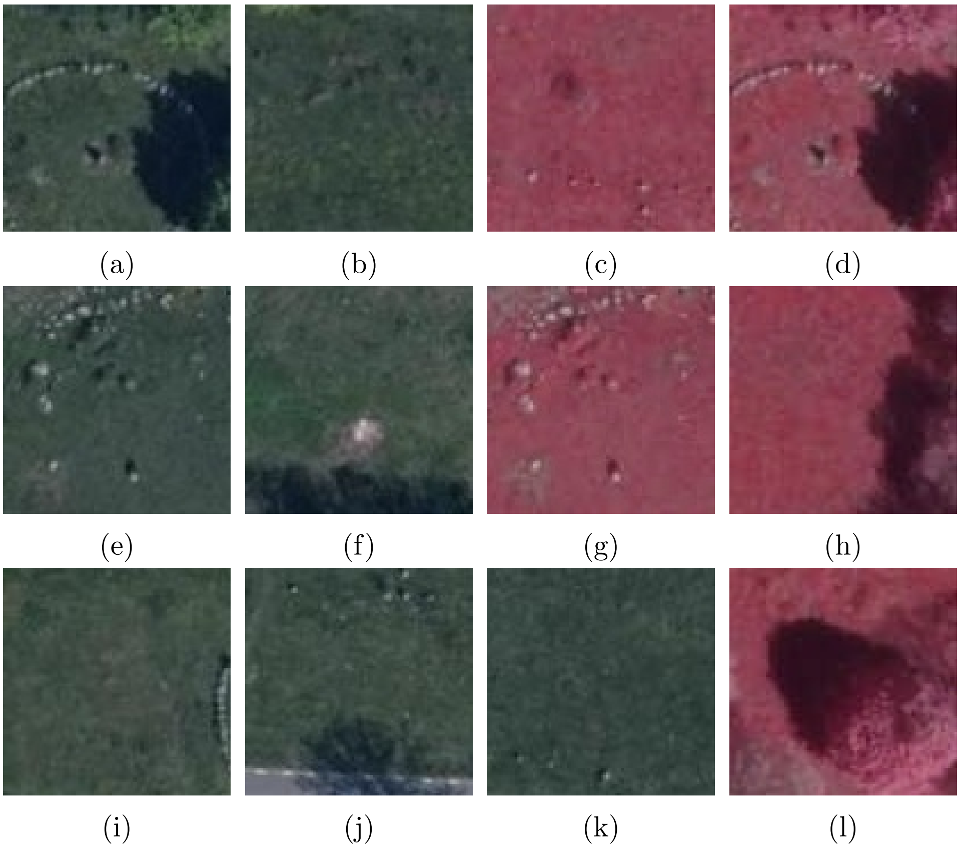


Figure 4.4: Some example of predictions, (a)-(d) true positive predictions, (e)-(h) false positives, (i)-(l) false negatives. ©Lantmäteriet: I2018 / 00119.

From the final outputs we can say that by combining different predictions results from different airborne data, in this case images in the infrared and visible spectrum, it is possible to reach an acceptable percentage of true detections. In the specific case study it also presents a very low percentage of false positives, having a maximum value of 8% with the best ensemble, a number which outperforms the studies which have been carried out using similar techniques [4, 45]. This improvemet in the performance can also be noticed from the ROC curves and AUC depicted in Figure 4.5, in four of the ensembles, the AUC is above 0.90 which is a better result than when the classifier perform predictions in a single wavelength. Specifically it is possible to say that the combination of the models VGG16 for the detection in the infrared, with Inception V3 for the detection in the visible light range, achieves a sensitivity of 76% and a specificity of 96%. This suggest that the combination of different models for different airborne data is the way in which the detection of archaeological should be implemented.

As it is possible to see, still some false positives are present even by using this method. Following the steps presented in Figure 1.1 a investigation on location was carried to visualize with precision what was causing these results. As reference, the system with the overall best performance shown in Figure 4.6 was taken. From the images of the exploration in the location we can see that some of the structures detected as false positives are indeed natural structures which are very similar to the structures of the graves, being these sets of rocks organized in a kind of geometrical shapes (Figure 4.9(a)-(b)). In some other cases, this corresponds to mounds inherent to the terrain (Figure 4.9(c)-(d)). For

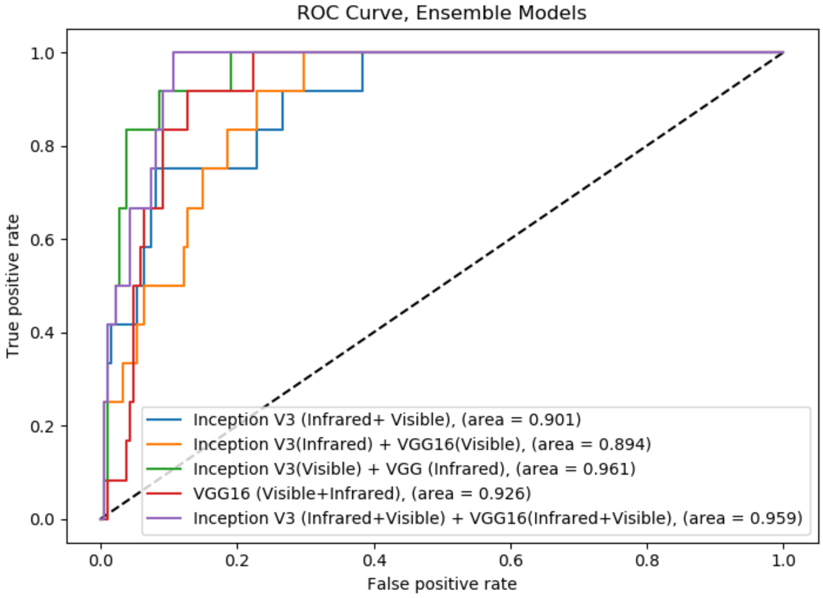


Figure 4.5: ROC curves obtained with the ensembles.

the false negatives it is a bit harder to try to determine why does the network fail to detect the complete structure as in the location it is also hard to recognize the archaeological structure, but at least a part of it is correctly detected (Figure 4.9(c)-(d)). If it is intended to develop a system which can deal with this controversial false positives, more data with this kind of images is needed so the network(s) can learn to distinguish the correct ones and reduce the number of these false positives.

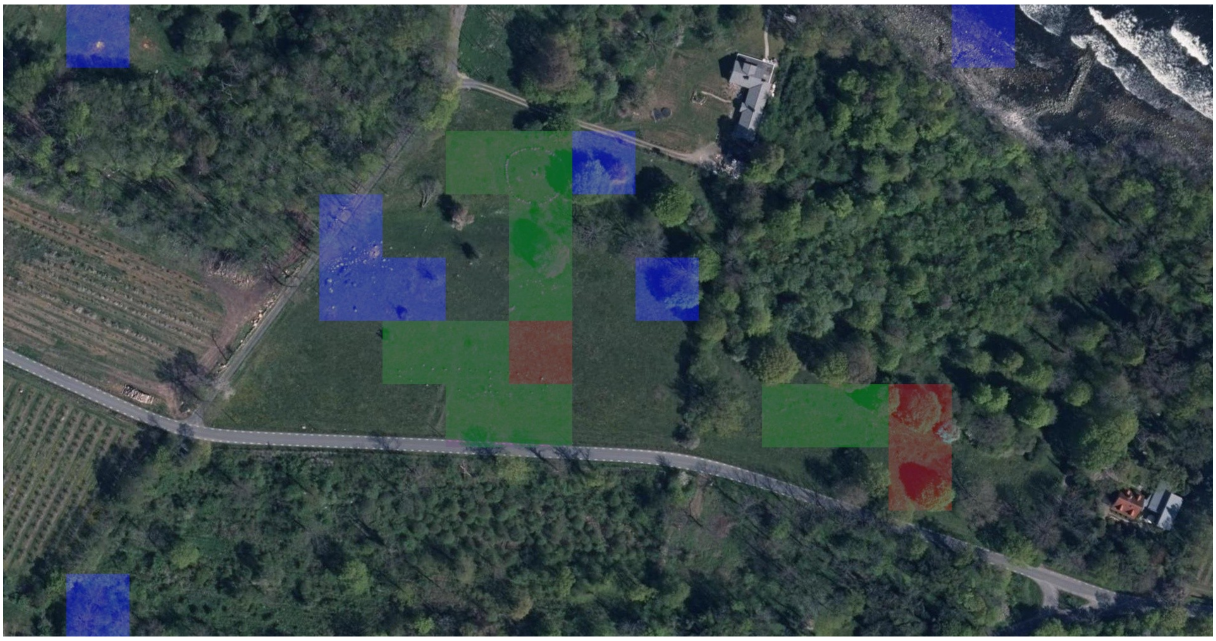


Figure 4.6: Combined results of the predictions using Inception V3 in the visible light data and VGG16 in the infrared. The tiles in green indicate correct detections, red missed detections and blue false positives. ©Lantmäteriet: I2018 / 00119.

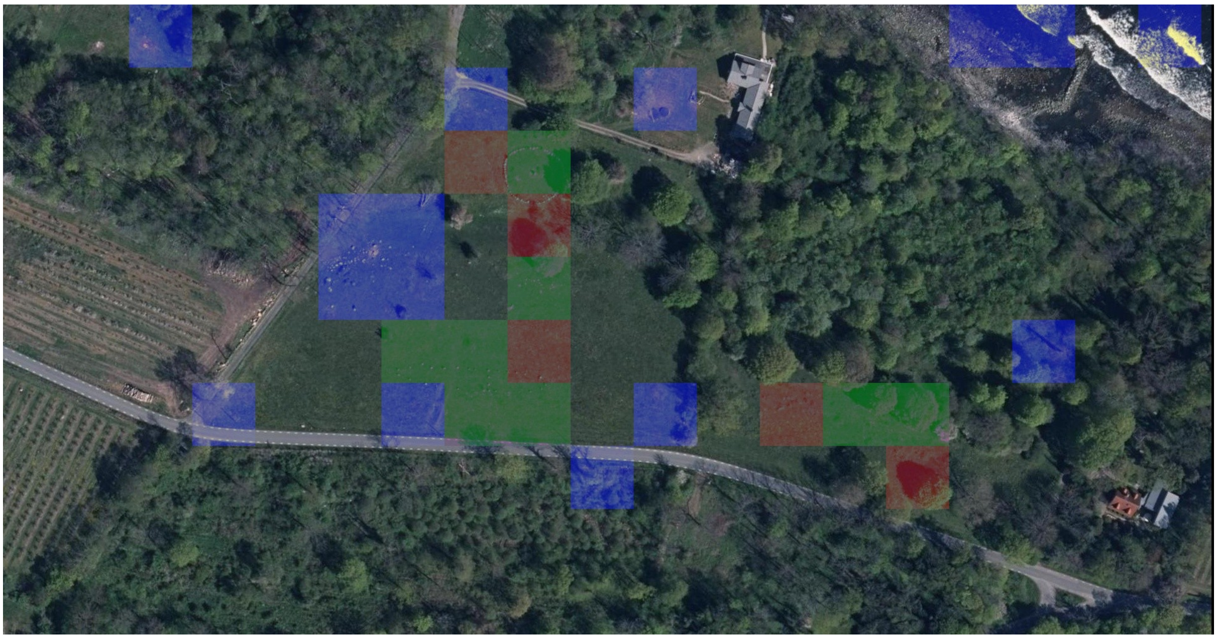


Figure 4.7: Combined results of the predictions in the infrared and visible light obtained using VGG16. The tiles in green indicate correct detections, red missed detections and blue false positives. ©Lantmäteriet: I2018 / 00119.

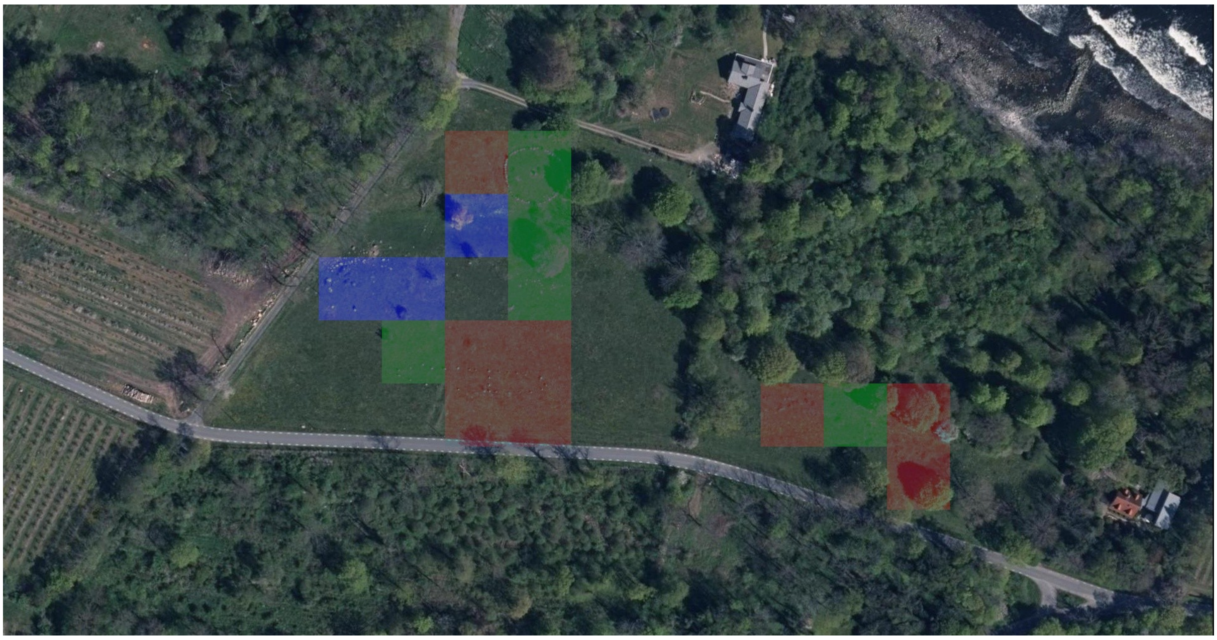
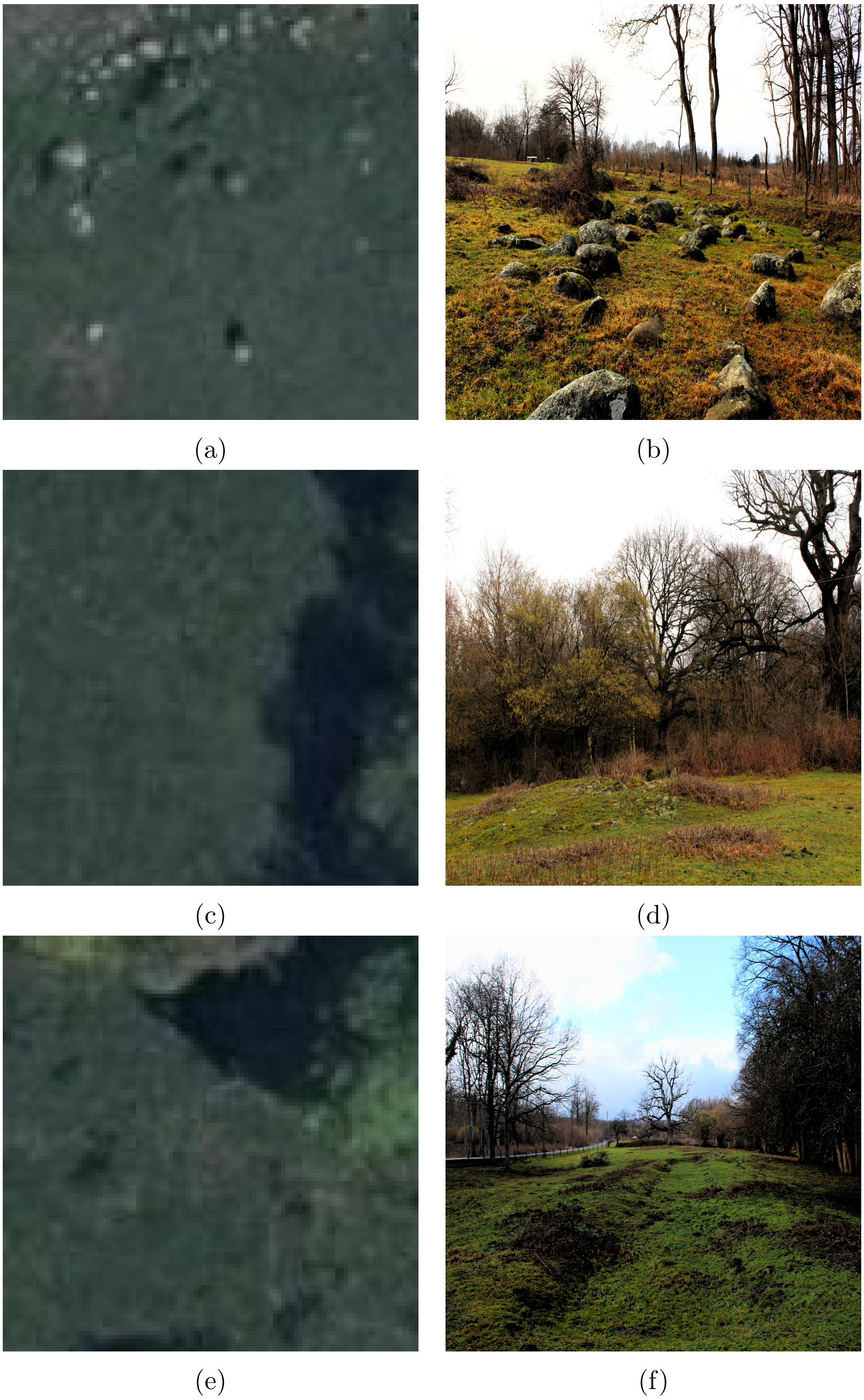


Figure 4.8: Combined results of the predictions in the infrared and visible light obtained using Inception V3. The tiles in green indicate correct detections, red missed detections and blue false positives. ©Lantmäteriet: I2018 / 00119.



(a) (b)

(c) (d)

(e) (f)

Figure 4.9: False detections corresponding to *controversial* images. Figures (a)-(d) correspond to false positives detected from the aerial images and its comparison at the location. Figures (e)-(f) are corresponding to a part of an structure which was not completely detected in all the tiles resulting in a false negative. (a),(c) & (e) ©Lantmäteriet: I2018 / 00119.

5

Summary and Conclusion

In this project 4 different networks were tested trying to address the task of detecting archaeological sites using airborne data. Due to the nature of the dataset, even if it contained different kinds of archaeological structures on it, it was decided that the classification task would determine if there is or not an archaeological trace in the images. For the same reason of having in total a small number of images with archaeological structures against the ones without, the ROC curve and the AUC were chosen as the performance measurement method during the parameter selection process. Three of the models tested were state of the art deep CNN architectures for which different techniques of transfer learning were tried, and the last one was a CNN model constructed from scratch. The pretrained models used were Inception V3, VGG16 and ResNet50, and for each of them, different parameters were adjusted in order to maximize the number of detections and reduce the number of false positives. The selection process was performed by doing 4-fold cross validation.

Once the models were tuned for this specific task, a comparison of the predictions obtained by training the systems in different datasets was carried out. The datasets tested were the original one and two others in which different data augmentation techniques were implemented. The first data augmentation method was done by using common image transformation techniques; the second one was done by adding samples of images with archaeological traces generated using a DCGAN model.

These results were compared in the test dataset for two different kinds of aerial imagery, infrared and visible light images, in order to see how the use of different data augmentation techniques could influence in the predictions of the models on different type of airborne data.

The models which showed the best performance in both, infrared and visible light were the ones based in VGG16 and Inception V3 architectures. For all of the pretrained models it was found that the best results when making predictions in the test dataset was achieved by using the systems as feature extractors and then adding a set of layers containing a pooling stage, and two fully connected layers. In the case of the proposed CNN it was found that a model with 5 convolutional blocks has the best results. The network reached values of AUC over 0.91 in the validation dataset, however in the test dataset its performance decreased to a value of 0.86 in the best case.

By varying the threshold value of the predictions obtained in the test dataset it was found that the two models which in general performed better than the others, were good at detecting, with a high degree of accuracy, images which do not have archaeological structures. A deeper analysis of this, showed that the use of DCGAN model as data augmentation technique increased slightly the amount of true detections but also increased the amount of false positives, which may be related to the quality of the images generated. For all the cases the use of typical data augmentation techniques improved the amount of detections of images with archaeological traces substantially while diminishing the amount of false positives. Specifically it was found that the use of this kind of data augmentation was particularly good to improve the performance of detections using VGG16 in the infrared and Inception V3 in the visible light range.

Finally, it was found that the detection of archaeological sites from aerial imagery improves substantially when the predictions obtained from different type of data, in this case infrared and visible light images, were combined. Specifically it was found that the combination of the VGG16 model for the detections in the infrared within Inception V3 model in the visible light range achieves a rate of true positive detections of 76 % with a rate of false positives of 4 % which outperforms the previous studies which have used similar methods.

The results obtained in this work suggest that the combination of different models applied to different types of data is the best technique for this kind of application. The use of this method can reduce the rate of false positives and still keep a good rate of true detections, dealing with one of the main issues exposed by the archaeological community, which is the high numbers of false positive detections.

In this project different CNN architectures were tried and each of them showed different performances. Even if the pretrained models showed a better performance in the test dataset than the model developed for this specific task this opens the possibility to the creation of deeper models specialized for this kind of duties. The alternative of creating specialized CNN could be enhanced by the creation of an archaeological benchmark database in which different models could be trained and tested; at the same time this could boost the adoption of this kind of techniques by the archaeologist community.

It was also shown that the alternative of using a DCGAN model to generate aerial images with archaeological structures is possible. Even if the results with this kind of technique were not exceptional, this opens the door to carry more detailed investigations in this area. Furthermore, GAN based models have more applications than the generation of images. An interesting approach would be the use of GANs for image to image translation purposes. A hypothetical case could be the generation of training data for regions with an specific kind of terrain, in which the real locations of archaeological sites is scarce, from another site with similar archaeological structures but different kind of ground.

With the completion of this work, it possible to assert that the detection of archaeological structures from aerial imagery using deep learning may not be a trivial task, but it is possible.

Acknowledgements

I want to thank my supervisor, Mattias, for giving me the chance of working with such an interesting and original topic and also for his guidance and advise during the development of this project. To Nicolo and Amanda from the department Archaeology and Ancient History, for providing me with the data and all the advise and information that me, a non-archaeologist, needed to carry out this task.

To the CONACYT, which sponsored this opportunity to study at Lund University.

To my friends, scattered all over the world, for their help and support.

Finally I want to thank my parents and my sister who have always been by my side, even in the distance.

Appendix A

Architectures

In this section a brief description of the pretrained models used is presented.

A.1 VGG16

VGG16 [75] is a 16-layers CNN model which was developed by the Visual Geometry Group from the Department of Engineering Science of Oxford University for the ILSVRC-2014 competition. The top-5 classification error achieved by the model was of 7.5 % on the validation set and 7.4 % on the test one, wining the first and the second places in the localisation and classification tracks respectively. The main characteristic of this network is that the parameters at each convolutional block are fixed, having all of them a convolutional filter of size 3x3. A graphical representation of it is depicted in figure A.1.

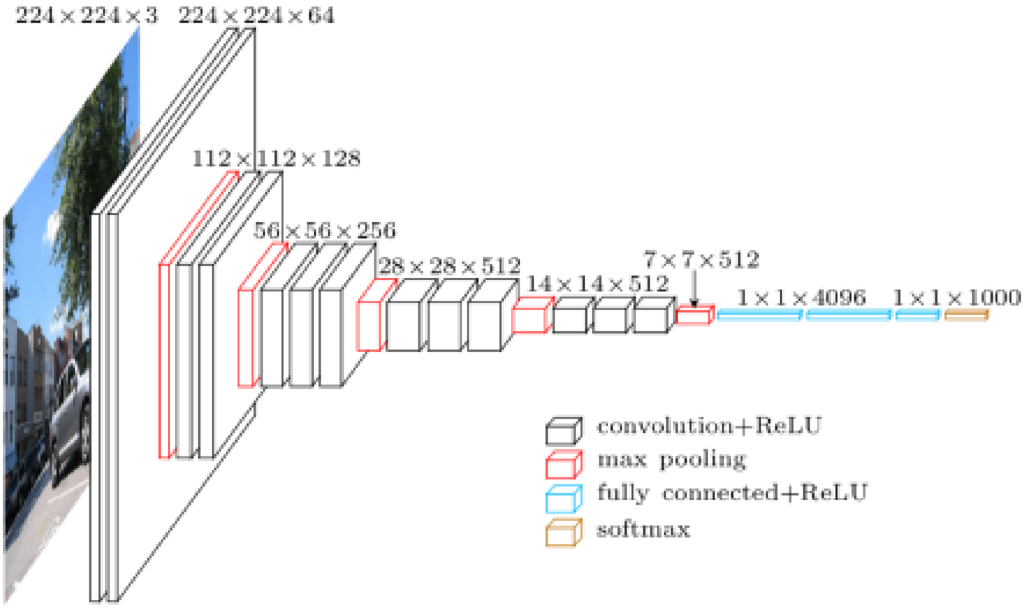


Figure A.1: Graphic representation of the VGG model.

A.2 Inception

Inception network can be considered a milestone in the way in which Deep CNN's models were developed. Before it, the standard way in which most of CNN's were build was by stacking convolutional layers, aiming to obtain better performances. One of the main advantajes of this model is that it requires abut 3 times less parameters in comparison to VGG16, therefore the computational cost for training this network is also less.

The proposal presented in [74] makes use of different convolutional sizes in parallel which previously, pass through a 1x1 convolution, this step has two main purposes: adding more non-linearity by having a ReLu operation just after every 1x1 convolution and also to reduce the dimensions inside the inception module, the output of each convolution is concatenated before passing to the next module as depicted in figure A.2.

A difference between its processors V1 and V2 is that it also makes use of Batch Normalization to the network and it uses of RMSProp Optimizer.

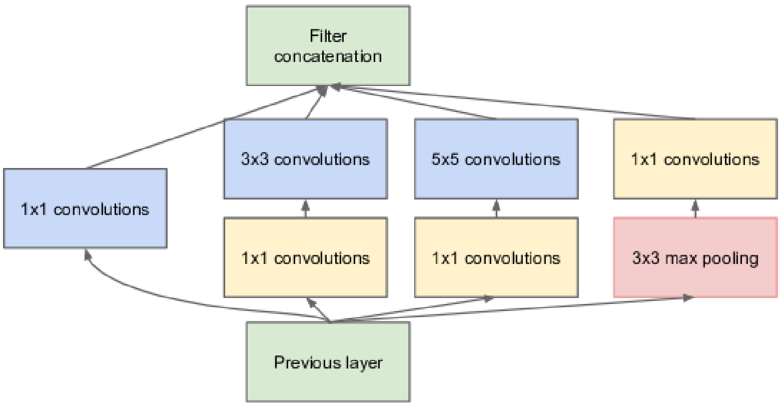


Figure A.2: Inception Module

A.3 ResNet50

Residual Networks, introduced in [76] and developed by Microsoft Research present a change in the paradigm of how to construct CNN models. This kind of model won the 1st place in the LSVRC 2015 classification competition with top-5 error rate of 3.57. The core idea behind this models is the use of Residual Blocks.

In this type of networks a mapping $H(x)$ is fitted in another underlying mapping $F(x) = H(x)x$ as it is shown in figure A.3. The core idea is that multiple non-linear layers can approximate the residual function $F(x)$. The connection of the input to the output is called an identity mapping. Residual learning is implemented to every stacked layer. Considering the building block as

$$y = F(x, \{W_i\}) + x \tag{A.3.1}$$

With $F(x, \{W_i\})$ the residual map to be learned. The operation $F + x$ is performed by shortcut connection and the addition is followed by an activation function σ. The addition between feature maps requires the output of the input layer of the residual block to be the same size as the one of the last layer of the residual block.

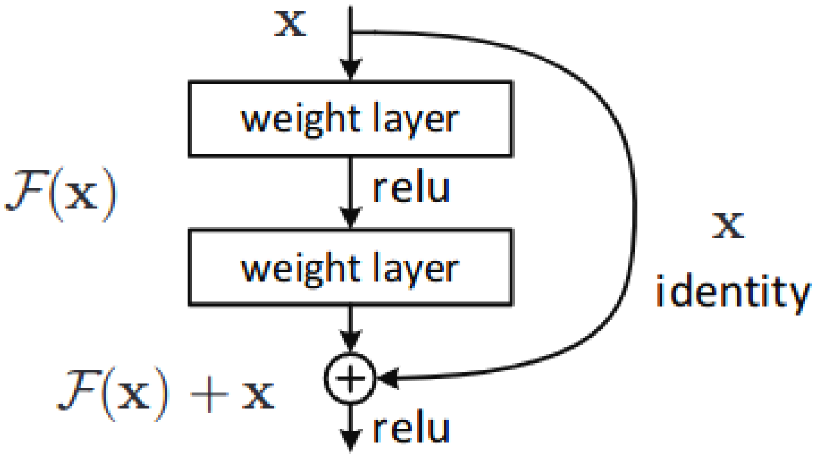


Figure A.3: Residual block. Image source: [79]

Appendix B

Previous studies in detection of archaeological sites using machine learning.

Method	Objects to detect	TPR	FPR	Location of the data used
Pre-trained CNN (Alexnet) + SVM [47]	Charcoal burning platforms	70%	72%	Valley of Lesja, Oppland County, Norway.
Pre-trained CNN (ResNet18) [4]	Roundhouses, Shielling huts and Small cairns	39%	122%	Island of Arran, Scotland.
Pre-trained CNN's tested separately (AlexNet, VGG19, CaffeNet, GoogLeNet, OverFeat) [45]	Square structures	100%	124%	Silvretta Alps, Switzerland.

Table B.1: Studies in which machine learning techniques have been used for archaeological object identification in aerial imagery

Appendix C

Images Obtained

In this appendix the images obtained by using different models in the test set for both, infrared and visible light are shown. As reference point, the image of the test area in visible light, is showed in figure C.1 where the tiles containing archaeological structures are coloured in purple.

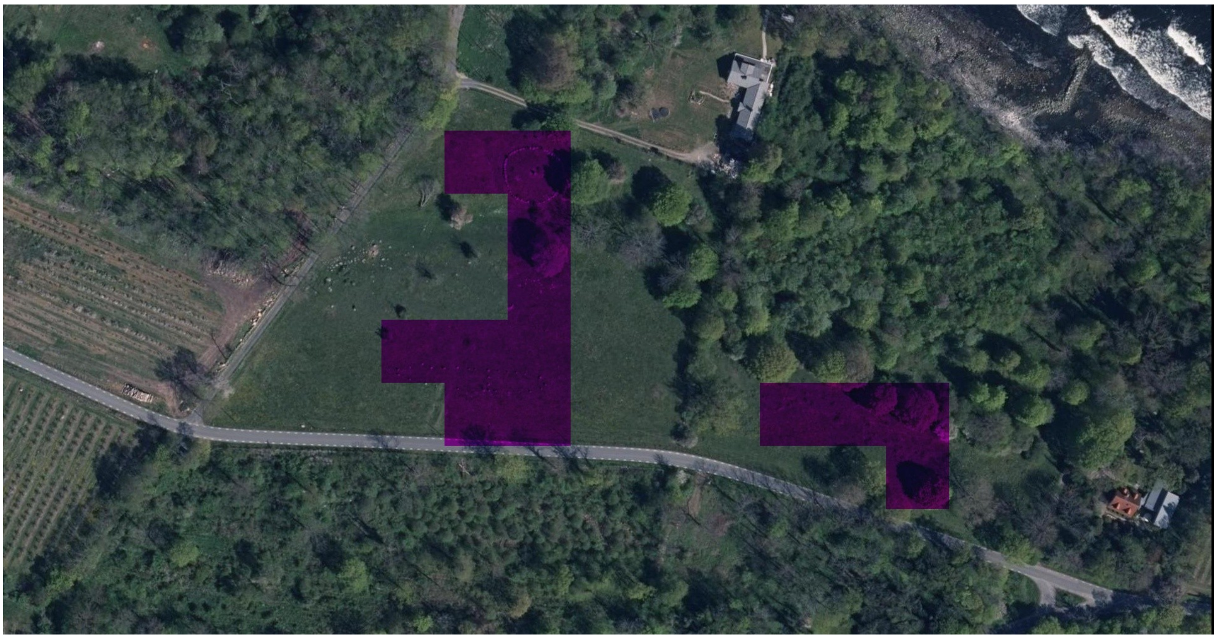
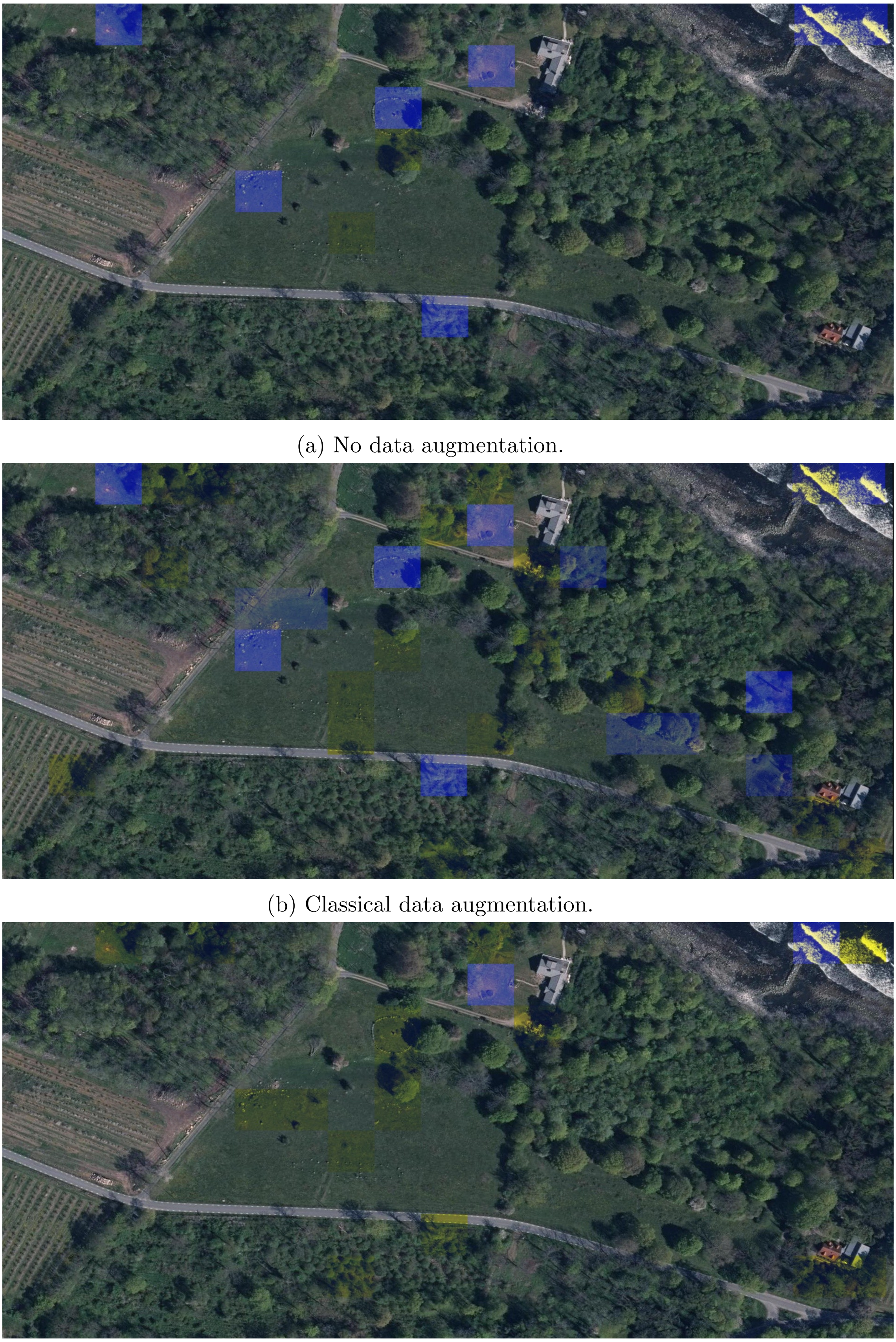


Figure C.1: Case study 4 in visible light with the tiles with archaeological traces marked in purple. ©Lantmäteriet: I2018 / 00119.

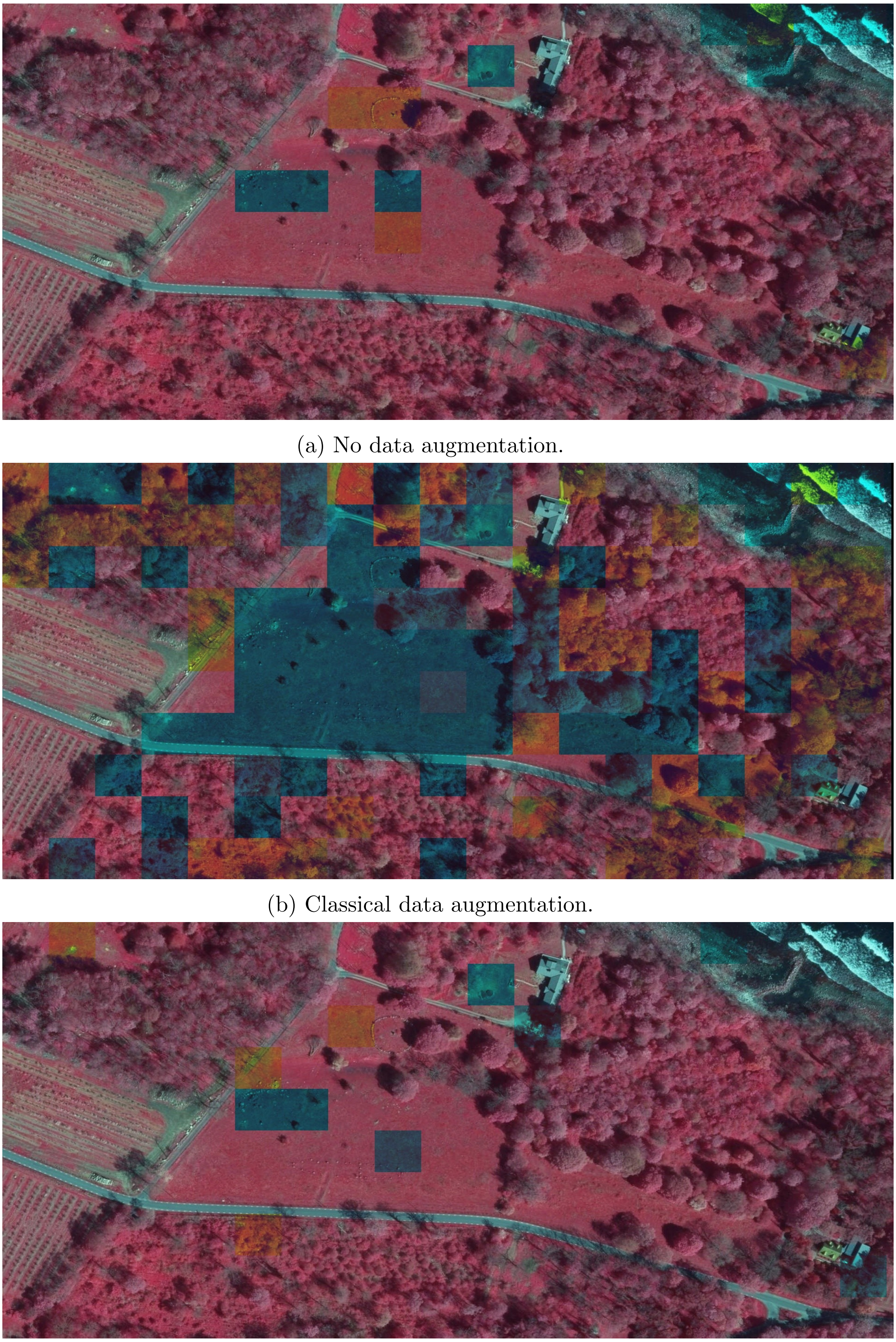


(a) No data augmentation.

(b) Classical data augmentation.

(c) Data augmentation using DCGAN method.

Figure C.2: Results obtained using VGG16 in aerial images in the visible light range for different data augmentation techniques. The tiles are colored according to the classification score for the positive class: blue > 0.5, light blue $\in [0.3, 0.5]$, yellow $\in [0.1, 0.3]$. ©Lantmäteriet: I2018 / 00119.

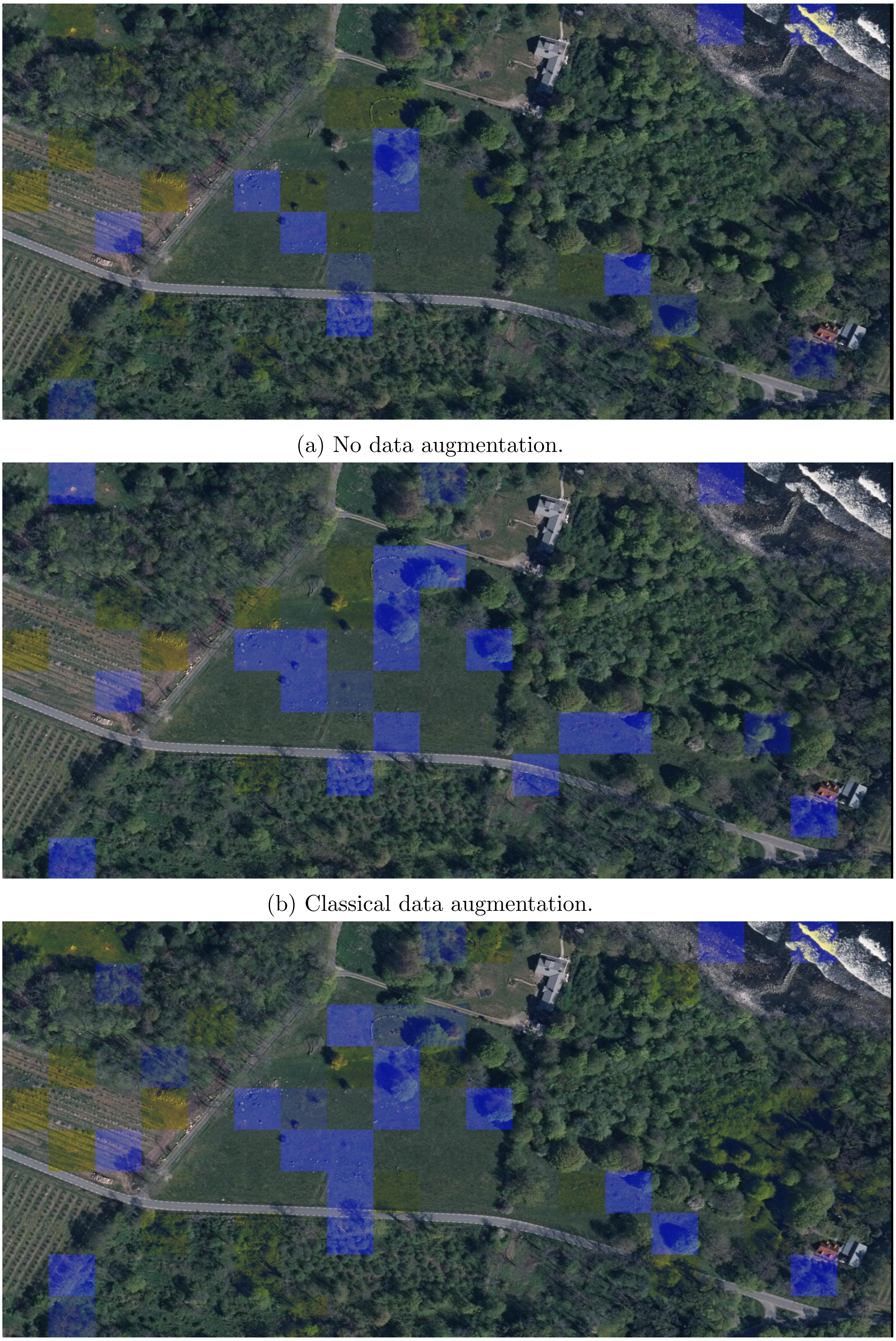


(a) No data augmentation.

(b) Classical data augmentation.

(c) Data augmentation using DCGAN method.

Figure C.3: Results obtained using VGG16 in aerial images in the infrared range for different data augmentation techniques. The tiles are colored according to the classification score for the positive class: blue > 0.5, light blue $\in [0.3, 0.5]$, yellow $\in [0.1, 0.3]$. ©Lantmäteriet: I2018 / 00119.

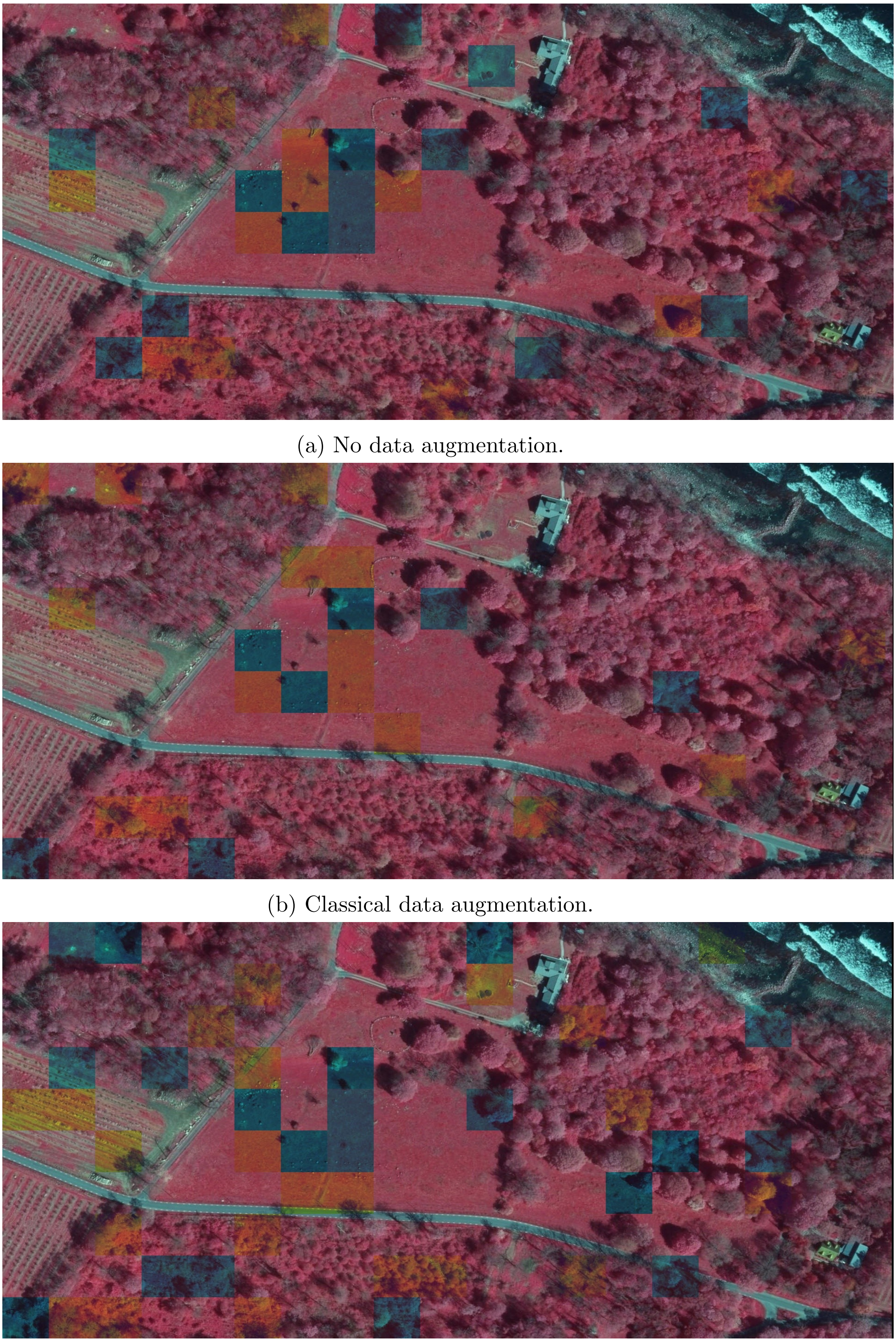


(a) No data augmentation.

(b) Classical data augmentation.

(c) Data augmentation using DCGAN method.

Figure C.4: Results obtained using Inception V3 in aerial images in the visible light range for different data augmentation techniques. The tiles are colored according to the classification score for the positive class: blue > 0.5, light blue $\in [0.3, 0.5]$, yellow $\in [0.1, 0.3]$. ©Lantmäteriet: I2018 / 00119.

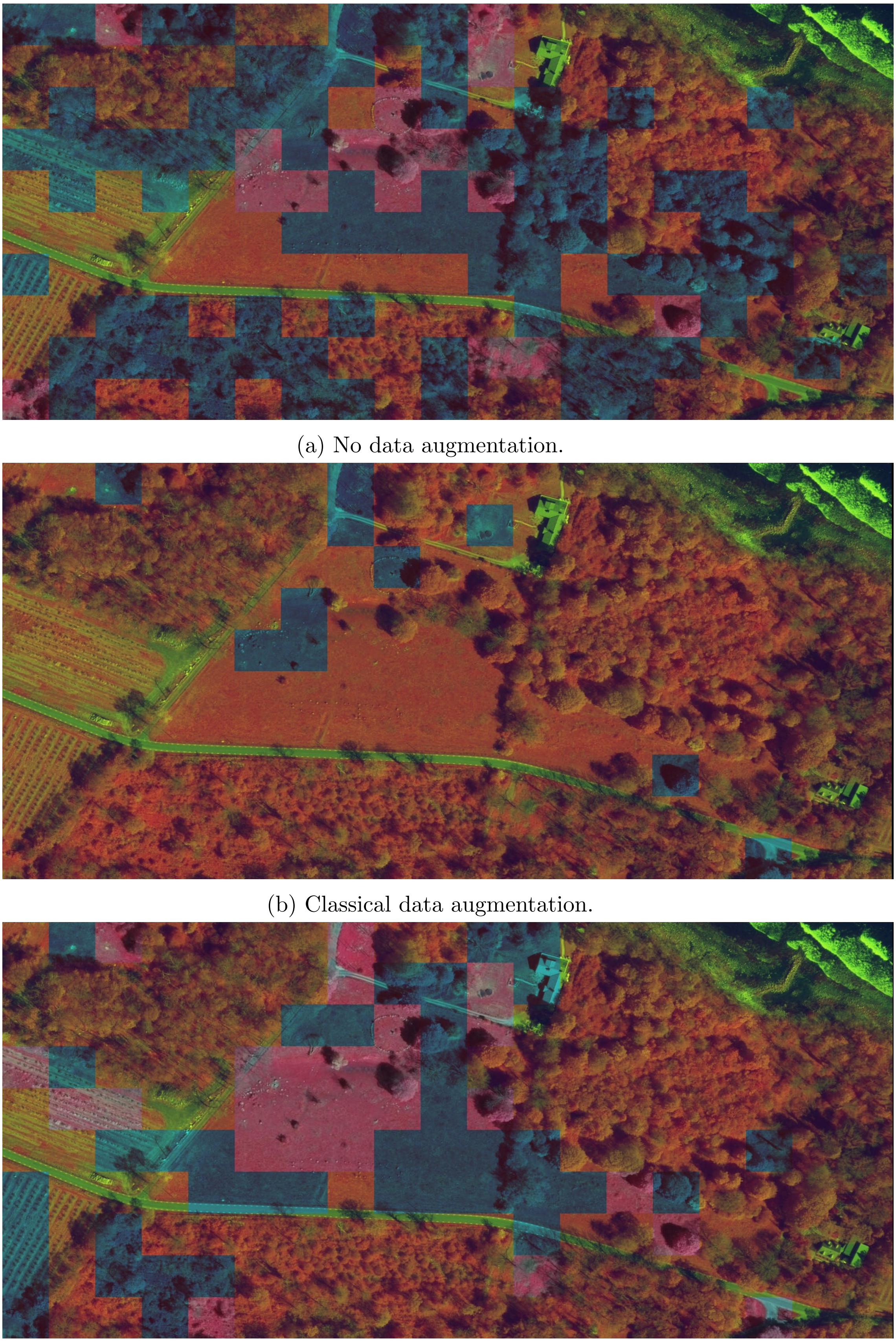


(a) No data augmentation.

(b) Classical data augmentation.

(c) Data augmentation using DCGAN method.

Figure C.5: Results obtained using Inception V3 in aerial images in the infrared range for different data augmentation techniques. The tiles are colored according to the classification score for the positive class: blue > 0.5, light blue $\in [0.3, 0.5]$, yellow $\in [0.1, 0.3]$. ©Lantmäteriet: I2018 / 00119.

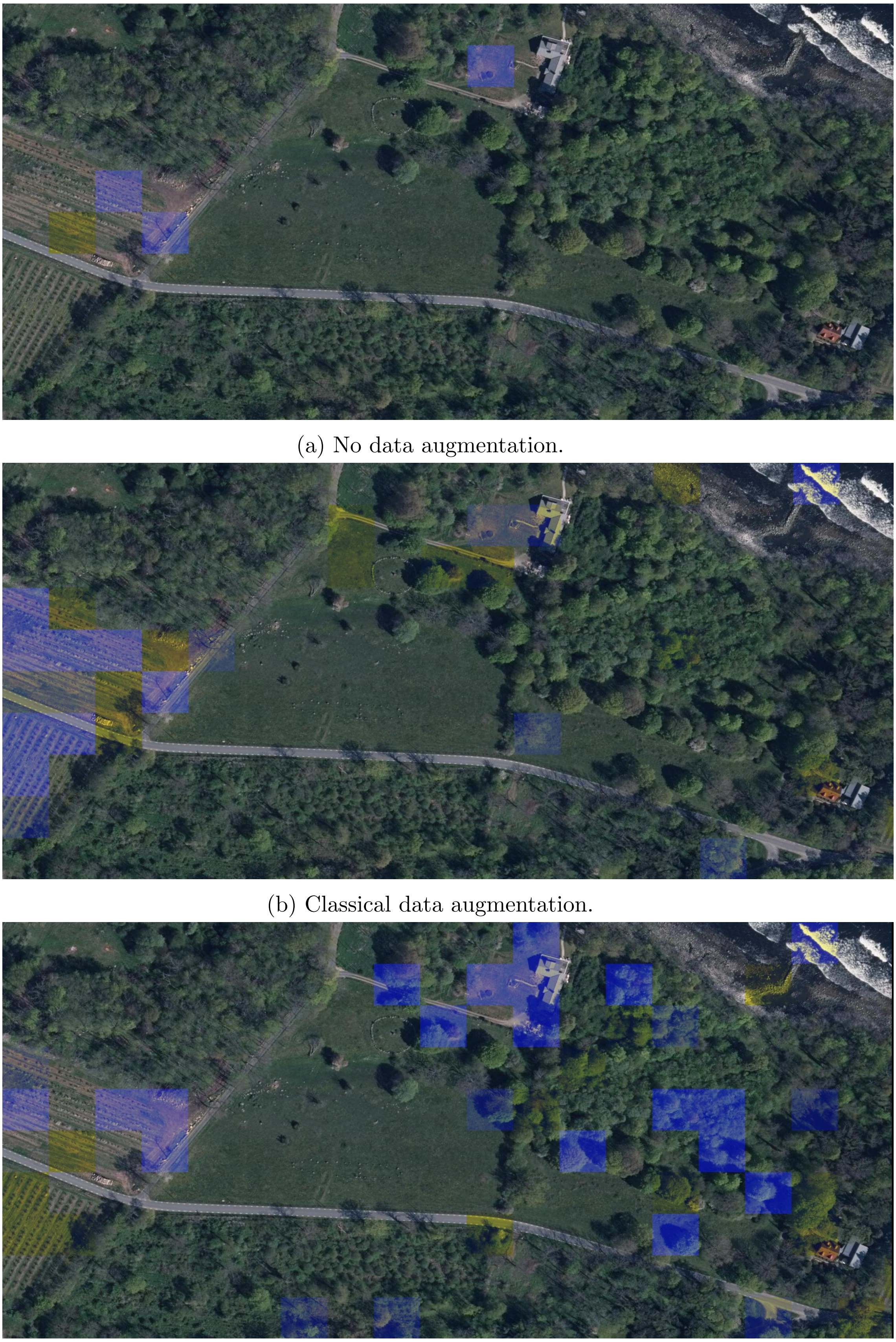


(a) No data augmentation.

(b) Classical data augmentation.

(c) Data augmentation using DCGAN method.

Figure C.6: Results obtained using ResNet50 in aerial images in the infrared range for different data augmentation techniques. The tiles are colored according to the classification score for the positive class: blue > 0.5, light blue $\in [0.3, 0.5]$, yellow $\in [0.1, 0.3]$. ©Lantmäteriet: I2018 / 00119.



(a) No data augmentation.

(b) Classical data augmentation.

(c) Data augmentation using DCGAN method.

Figure C.7: Results obtained using the proposed CNN in aerial images in the visible light range for different data augmentation techniques. The tiles are colored according to the classification score for the positive class: blue > 0.5, light blue $\in [0.3, 0.5]$, yellow $\in [0.1, 0.3]$. ©Lantmäteriet: I2018 / 00119.

Appendix D

Images generated using a Deep Generative Adversarial Networks

The Deep Generative adversarial model used was based on [72], the training was carried using only the original training dataset of positives images, i.e. 82 images in total for training. The system was trained for 250 epochs obtaining a final loss of 0.004274 for the discriminator and 0.56293 for the generator. The system in the last epoch reached an accuracy of 0.9997. Some of the images generated are shown in figure D.1

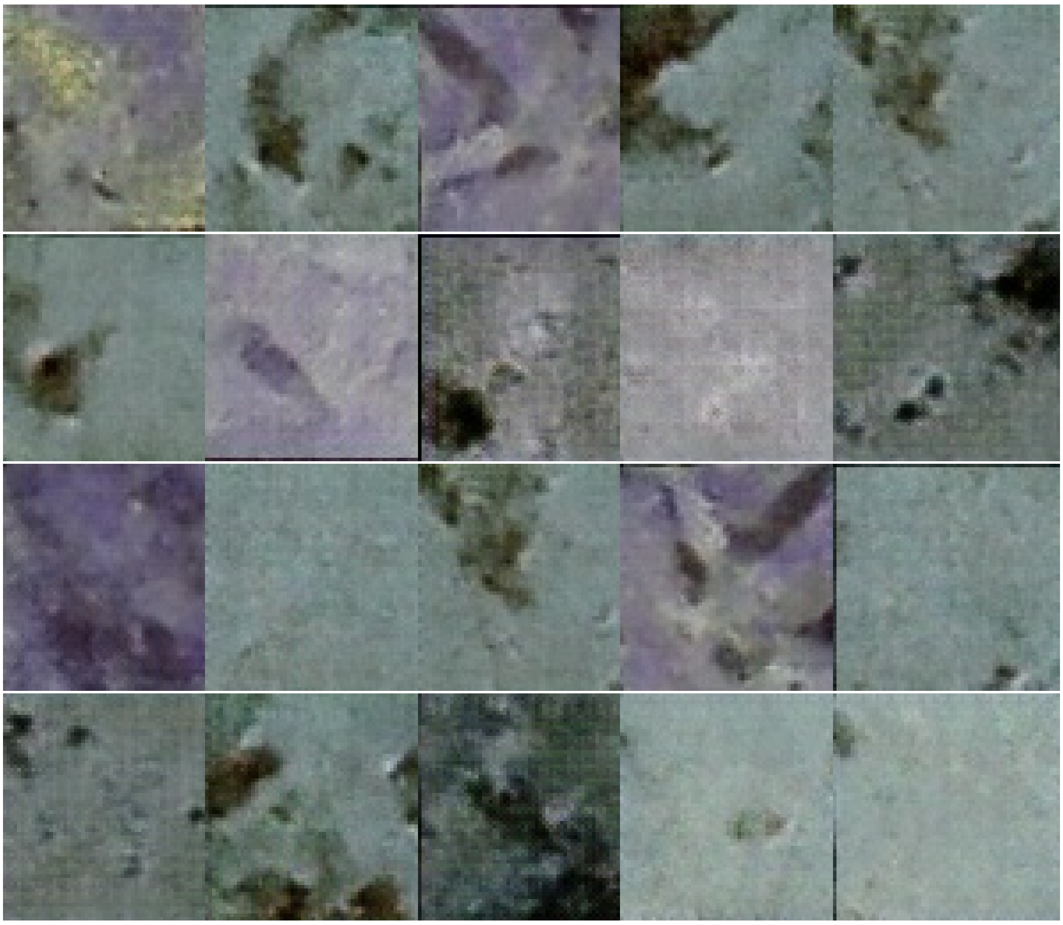


Figure D.1: Some samples of the images generated using DGANs

Bibliography

[1] van der Maaten L, Boon P, Lange G. Computer Vision and Machine Learning for Archaeology; 2014.

[2] Chetouani A, Debroutelle T, Treuillet S, Exbrayat M, Jesset S. Classification of Ceramic Shards Based on Convolutional Neural Network. In: 2018 25th IEEE International Conference on Image Processing (ICIP); 2018. p. 1038–1042.

[3] Menze BH, Ur JA, Sherratt AG. Detection of Ancient Settlement Mounds: Archaeological Survey Based on the SRTM Terrain Model. Photogrammetric Engineering Remote Sensing. 2006 March;72(3):321–327.

[4] Trier , Cowley DC, Waldeland AU. Using deep neural networks on airborne laser scanning data: Results from a case study of semi-automatic mapping of archaeological topography on Arran, Scotland. Archaeological Prospection;0(0). Available from: `https://onlinelibrary.wiley.com/doi/abs/10.1002/arp.1731`.

[5] Goodfellow I, Bengio Y, Courville A. Deep Learning. New York: MIT Press; 2016. `http://www.deeplearningbook.org`.

[6] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015 5;521(7553):436–444.

[7] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25. Curran Associates, Inc.; 2012. p. 1097–1105. Available from: `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

[8] Lake BM, Salakhutdinov R, Tenenbaum JB. Human-level concept learning through probabilistic program induction. Science. 2015;350(6266):1332–1338. Available from: `http://science.sciencemag.org/content/350/6266/1332`.

[9] Mnih V. Machine Learning for Aerial Image Labeling; 2013.

[10] AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification;. Accessed: 2018-12-21. `http://captain.whu.edu.cn/WUDA-RSImg/aid.html`.

[11] Hu F, Xia GS, Hu J, Zhang L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. Remote Sensing. 2015;7(11):14680–14707. Available from: `http://www.mdpi.com/2072-4292/7/11/14680`.

[12] Castelluccio M, Poggi G, Sansone C, Verdoliva L. Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. CoRR. 2015;abs/1508.00092. Available from: http://arxiv.org/abs/1508.00092.

[13] Penatti OAB, Nogueira K, dos Santos JA. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: CVPR Workshops. IEEE Computer Society; 2015. p. 44–51. Available from: http://dblp.uni-trier.de/db/conf/cvpr/cvprw2015.html#PenattiNS15.

[14] Xia GS, Yang W, Delon J, Gousseau Y, Sun HPH, Maıtre H. Structural High-resolution Satellite Image Indexing; 2010. .

[15] Azimi SM, Fischer P, Körner M, Reinartz P. Aerial LaneNet: Lane Marking Semantic Segmentation in Aerial Imagery using Wavelet-Enhanced Cost-sensitive Symmetric Fully Convolutional Neural Networks; 2018.

[16] Khryashchev V, Pavlov V, Priorov A, Kazina E. Convolutional Neural Network for Satellite Imagery;.

[17] Mokhtarzade M, Zoej MJV. Road detection from high-resolution satellite images using artificial neural networks; 2016.

[18] Gleason CJ, Im J. Forest biomass estimation from airborne LiDAR data using machine learning approaches. Remote Sensing of Environment. 2012;125:80 – 91. Available from: http://www.sciencedirect.com/science/article/pii/S0034425712002787.

[19] Vetrivel A, Gerke M, Kerle N, Nex F, Vosselman G. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. ISPRS Journal of Photogrammetry and Remote Sensing. 2018;140:45 – 59. Geospatial Computer Vision. Available from: http://www.sciencedirect.com/science/article/pii/S0924271616305913.

[20] Jean1 N, Burke M, Xie M, Davis WM, Lobell DB, Ermon S. Combining satellite imagery and machine learning to predict poverty. 2016;5:790–794.

[21] Xia GS, Hu J, Hu F, Shi B, Bai X, Zhong Y, et al. AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. IEEE Transactions on Geoscience and Remote Sensing. 2017;55:3965–3981.

[22] Nogueira K, Penatti OAB, dos Santos JA. Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification. CoRR. 2016;abs/1602.01517. Available from: http://arxiv.org/abs/1602.01517.

[23] UC Merced Land Use Dataset;. Accessed: 2018-12-21. http://weegee.vision.ucmerced.edu/datasets/landuse.html.

[24] Vaishnnave MP, andP Srinivasan KSD, Jothi GAP. A Review on Deep Learning Neural Network Datasets for Satellite Imagery. International Journal for Research in Applied Science Engineering Technology (IJRASET). 2018;6. Available from: https://www.ijraset.com/fileserve.php?FID=19086.

[25] Rebecca Bennett VDL Dave Cowley. The data explosion: tackling the taboo of automatic feature recognition in airborne survey data;.

[26] Traviglia A, Cowley D, Lambers K. Finding common ground: human and computer vision in archaeological prospection. AARGnews - The newsletter of the Aerial Archaeology Research Group. 2016;53:14. Available from: https://openaccess.leidenuniv.nl/handle/1887/43751.

[27] Figueredo AJ, Wolf PSA. A Future Perspective for Automated Detection of Archaeology using Deep Learning with Remote Sensor Data. Conference Paper, New Forest Knowledge Conference 2017. 2017;.

[28] History of Aerial Photography;. Accessed: 2018-12-27. https://papa.clubexpress.com/content.aspx?page_id=22&club_id=808138&module_id=158950.

[29] Ceraudo G. Aerial Photography in Archaeology. In: Good Practice in Archaeological Diagnostics. Non-invasive Survey of Complex Archaeological Sites. Springer; 2013. p. 266–290.

[30] Chase AF, Chase DZ, Awe JJ, Weishampel JF, Iannone G, Moyes H, et al. Ancient Maya Regional Settlement and Inter-Site Analysis: The 2013 West-Central Belize LiDAR Survey. Remote Sensing. 2014;6(9):8671–8695. Available from: http://www.mdpi.com/2072-4292/6/9/8671.

[31] Canuto MA, Estrada-Belli F, Garrison TG, Houston SD, Acuña MJ, Kováč M, et al. Ancient lowland Maya complexity as revealed by airborne laser scanning of northern Guatemala. Science. 2018;361(6409). Available from: http://science.sciencemag.org/content/361/6409/eaau0137.

[32] Chase AF, Chase DZ, Awe JJ, Weishampel JF, Iannone G, Moyes H, et al. The Use of LiDAR in Understanding the Ancient Maya Landscape: Caracol and Western Belize. Advances in Archaeological Practice. 2014;2(3):208–221.

[33] How Do Archaeological Sites Show;. Accessed: 2018-12-27. https://luftbildarchiv.univie.ac.at/aerial-archaeology/introduction-to-aerial-archaeology/visibility-marks/.

[34] L' ARCHEOLOGIE AERIENNE;. Accessed: 2018-12-27. http://archaero.com/Arch%E9ologie-a%E9rienne.htm.

[35] Aerial Photographs of the Crop Marks in the University Parks;. Accessed: 2018-12-27. http://users.ox.ac.uk/~parks/crops/index.htm.

[36] Sparavigna AC. The cradle of pyramids in satellite images. arXiv e-prints. 2011 Jun;p. arXiv:1106.0818.

[37] Miller G. Space Archaeology; 2016. Accessed: 2018-12-27. http://possibility.teledyneimaging.com/space-archaeology/.

[38] Doneus M, Verhoeven G, Fera M, Ch Briese MK, Neubauer W. From Deposit to Point Cloud – a Study of Low-Cost Computer Vision Approaches for the Straightforward Documentation of Archaeological Excavations. Geoinformatics FCE CTU. 2006;6. Available from: https://ojs.cvut.cz/ojs/index.php/gi/article/view/2636.

[39] ASTER Satellite Sensor;. Accessed: 2018-12-27. https://www.satimagingcorp.com/satellite-sensors/other-satellite-sensors/aster/.

[40] Archaeological Site Prediction Using Machine Learning; 2017. Accessed: 2018-12-27. https://sflscientific.com/data-science-blog/2017/1/9/archaeological-site-prediction-using-machine-learning.

[41] Menze BH, Ur JA. Mapping patterns of long-term settlement in Northern Mesopotamia at a large scale. Proceedings of the National Academy of Sciences. 2012;109(14):E778–E787. Available from: https://www.pnas.org/content/109/14/E778.

[42] Khabur Sites;. Accessed: 2018-12-27. https://fusiontables.googleusercontent.com/embedviz?q=select+col2+from+1i4jhkVuvvW60_sHEpb84b7KV2_dW8dBoDUnVhuY&viz=MAP&h=false&lat=36.733915793847046&lng=40.615469018554684&t=3&z=14&l=col2.

[43] Menze BH, Ur JA. Multitemporal Fusion for the Detection of Static Spatial Patterns in Multispectral Satellite Images—With Application to Archaeological Survey. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2014 Aug;7(8):3513–3524. Available from: https://ieeexplore.ieee.org/document/6907965.

[44] Bowen EFW, Tofel BB, Parcak S, Granger R. Algorithmic Identification of Looted Archaeological Sites from Space. Frontiers in ICT. 2017;4:4. Available from: https://www.frontiersin.org/article/10.3389/fict.2017.00004.

[45] Zingman I, Saupe D, Penatti OAB, Lambers K. Detection of Fragmented Rectangular Enclosures in Very-High-Resolution Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing. 2016;54(8):4580–4593.

[46] Kramer I, Hare J, Prugel-Bennett A. A Future Perspective for Automated Detection of Archaeology using Deep Learning with Remote Sensor Data.; 2017.

[47] Trier , Arnt-Borre S, Lars Holger P. Semi-automatic mapping of charcoal klins from airbone laser scanning data using deep learning. Oxford: Archaeopress;.

[48] Kaliff A. Grave Structures and Altars: Archaeological Traces of Bronze Age Eschatological Conceptions. European Journal of Archaeology. 1998;1(2):177–198. Available from: https://doi.org/10.1177/146195719800100203.

[49] Davidson HRE. The Road to Hel: A Study of the Conception of the Dead in Old Norse Literature. Greenwood Press; 1943. Available from: https://books.google.se/books?id=QllYngEACAAJ.

[50] Sawyer PH. Kings and Vikings : Scandinavia and Europe, A.D. 700-1100 / P.H. Sawyer. Methuen London ; New York; 1982.

[51] Anderson J. Notes on the Contents of two Viking Graves in Islay, Discovered by William Campbell, Esq., of Ballinaby; with Notices of the Burial Customs of the Norse Sea-Kings, as recorded in the Sagas and Illustrated by their Grave Mounds in Norway and in Scotland. Proceedings of the Society of Antiquaries of Scotland.

1880 Nov;14:51–89. Available from: http://journals.socantscot.org/index.php/psas/article/view/5902.

[52] Henriksson M, Landeschi G. Skadeinventering och undersökning av Hjortahammar gravfält; 2017. http://www.blekingemuseum.se/reports/307.

[53] Birka och Hovgården;. https://www.raa.se/evenemang-och-upplevelser/upplev-kulturarvet/varldsarv-i-sverige/birka-och-hovgarden/.

[54] Birka and Hovgården;. https://whc.unesco.org/en/list/555.

[55] QGIS. A Free and Open Source Geographic Information System;. Accessed: 2019-01-08. https://qgis.org/en/site/.

[56] How to Retrain an Image Classifier for New Categories;. Accessed: 2018-11-07. https://www.tensorflow.org/hub/tutorials/image_retraining.

[57] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems 25. Curran Associates, Inc.; 2012. p. 1097–1105. Available from: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[58] Shyamal P, Johanna P. Introduction to Deep Learning: What Are Convolutional Neural Networks?;. Accessed: 2018-12-19. https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--14895127657
html.

[59] Jost Tobias S, Alexey D, Thomas B, Martin R. Striving for Simplicity: The All Convolutional Net; 2015. https://arxiv.org/abs/1412.6806.

[60] Boureau Y, Ponce J, LeCun Y. A theoretical analysis of feature pooling in visual recognition. In: ICML 2010 - Proceedings, 27th International Conference on Machine Learning; 2010. p. 111–118.

[61] Notes of CS231n: Convolutional Neural Networks for Visual Recognition;. C. Available from: http://cs231n.github.io/.

[62] Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks?; 2014. https://arxiv.org/abs/1411.1792.

[63] Antoniou A, Storkey A, Edwards H. Data Augmentation Generative Adversarial Networks; 2018. https://arxiv.org/abs/1711.04340.

[64] Dosovitskiy A, Brox T. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, editors. Advances in Neural Information Processing Systems 29. Curran Associates, Inc.; 2016. p. 658–666. Available from: http://papers.nips.cc/paper/6158-generating-images-with-perceptual-similarity-metrics-based-on-deep-networks.pdf.

[65] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in Neural Information Processing Systems 27. Curran Associates, Inc.; 2014. p. 2672–2680. Available from: `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[66] Paganini M, de Oliveira L, Nachman B. Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters. . 2018 Jan;120:042003.

[67] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative Adversarial Networks: An Overview. IEEE Signal Processing Magazine. 2018 Jan;35(1):53–65.

[68] Zhai M, Bessinger Z, Workman S, Jacobs N. Predicting Ground-Level Scene Layout from Aerial Imagery. arXiv e-prints. 2016 Dec;p. arXiv:1612.02709.

[69] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-Image Translation with Conditional Adversarial Networks. arXiv e-prints. 2016 Nov;p. arXiv:1611.07004.

[70] Costea D, Marcu A, Slusanschi E, Leordeanu M. Creating Roadmaps in Aerial Images With Generative Adversarial Networks and Smoothing-Based Optimization. In: The IEEE International Conference on Computer Vision (ICCV) Workshops; 2017. .

[71] Yi Z, Zhang H, Tan P, Gong M. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. arXiv e-prints. 2017 Apr;p. arXiv:1704.02510.

[72] Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv e-prints. 2015 Nov;p. arXiv:1511.06434.

[73] Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters. 2006;27(8):861 – 874. ROC Analysis in Pattern Recognition. Available from: `http://www.sciencedirect.com/science/article/pii/S016786550500303X`.

[74] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the Inception Architecture for Computer Vision. arXiv e-prints. 2015 Dec;p. arXiv:1512.00567.

[75] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv e-prints. 2014 Sep;p. arXiv:1409.1556.

[76] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. arXiv e-prints. 2015 Dec;p. arXiv:1512.03385.

[77] ImageNet Overview;. Accessed: 2018-11-07. `http://image-net.org/about-overview`.

[78] Chetouani A, Debroutelle T, Treuillet S, Exbrayat M, Jesset S. Classification of Ceramic Shards Based on Convolutional Neural Network. In: 2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018; 2018. p. 1038–1042. Available from: `https://doi.org/10.1109/ICIP.2018.8451728`.

[79] Convolutional Neural Networks for Image and Video Processing;. Accessed: 2019-01-08. https://wiki.tum.de/display/lfdv/Deep+Residual+Networks.